



De la modélisation objet des logiciels à la metamodélisation des langages informatiques

pierre-alain.muller@irisa.fr

20 novembre 2006



CV Rapide

- ▶ 1988-1993 Consultant Rational Software
- ▶ 1994-1998 MC Université de Haute-Alsace
- ▶ 1999-2002 PDG Objexion Software
- ▶ 2003-2004 MC ESSAIM
- ▶ 2005-2006 Délégation INRIA
- ▶ Et maintenant MC ENSISA



Plan de la présentation

- ▶ Contexte
- ▶ Problématique
- ▶ Contributions
- ▶ Perspectives

Concepts

Représentation



Choses



Contexte





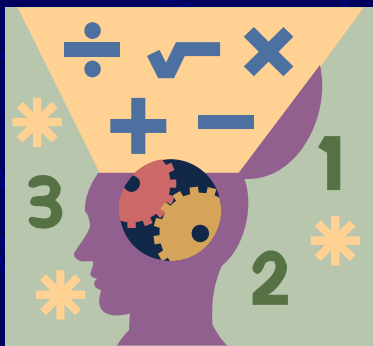
Les choses et leurs représentations

► Les choses

- Réelles, virtuelles
- Rares, chères, fragiles, dangereuses, inaccessibles, lointaines, trop nombreuses...

► Les concepts pour penser les choses

- Plus facile, moins cher, moins dangereux...

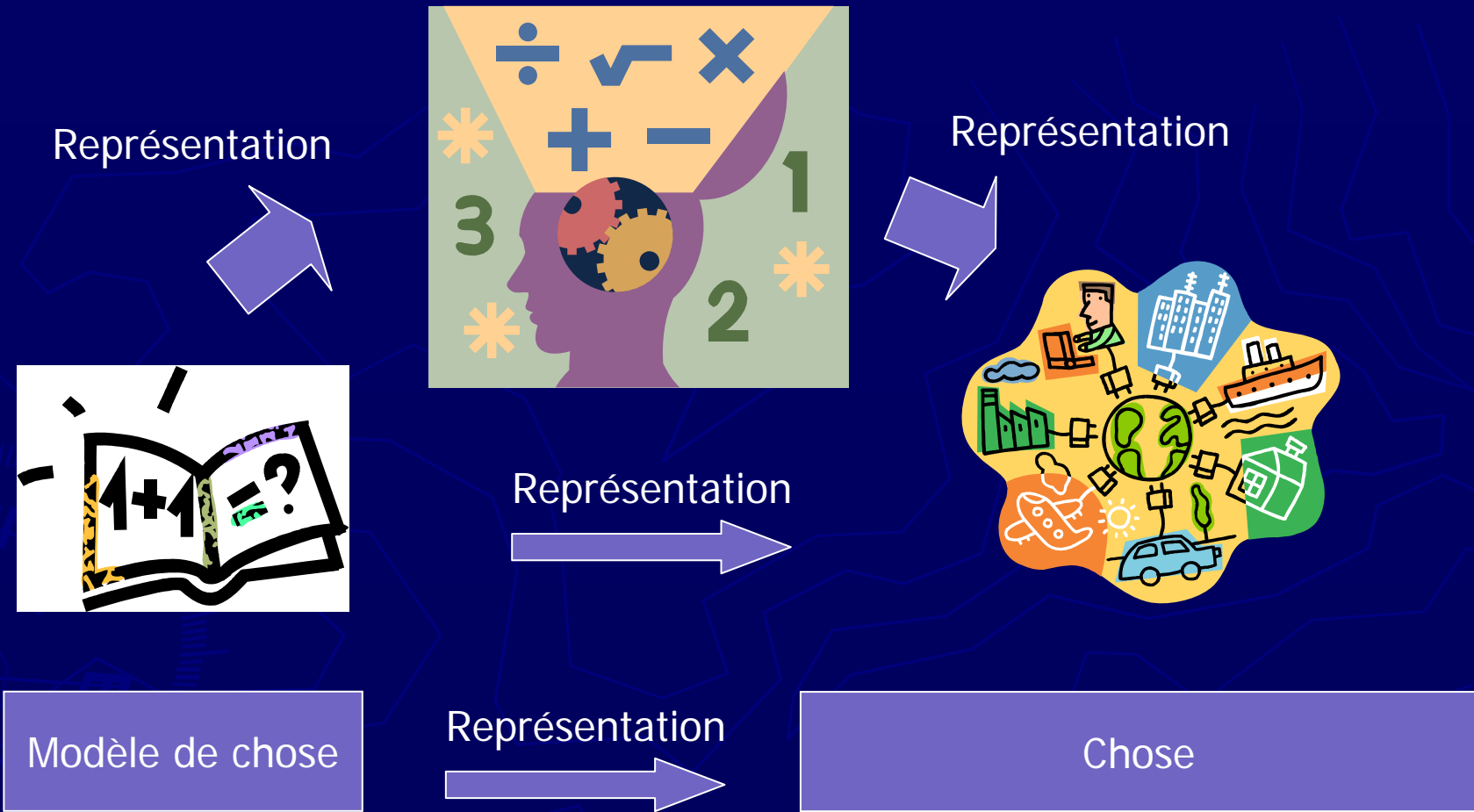


Représentation





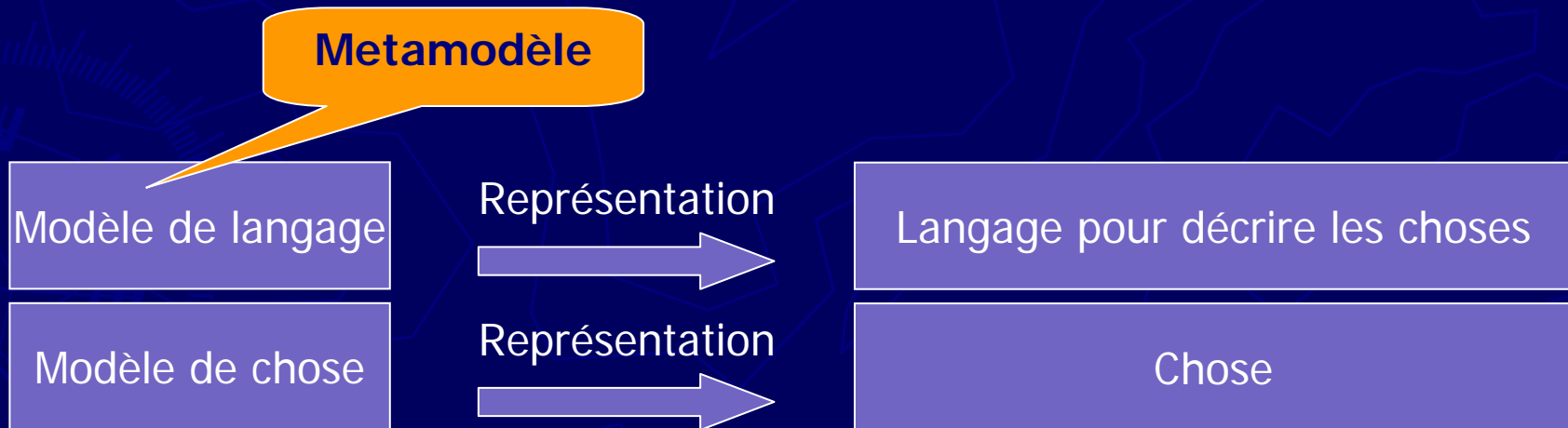
Représentation mentale vs. physique





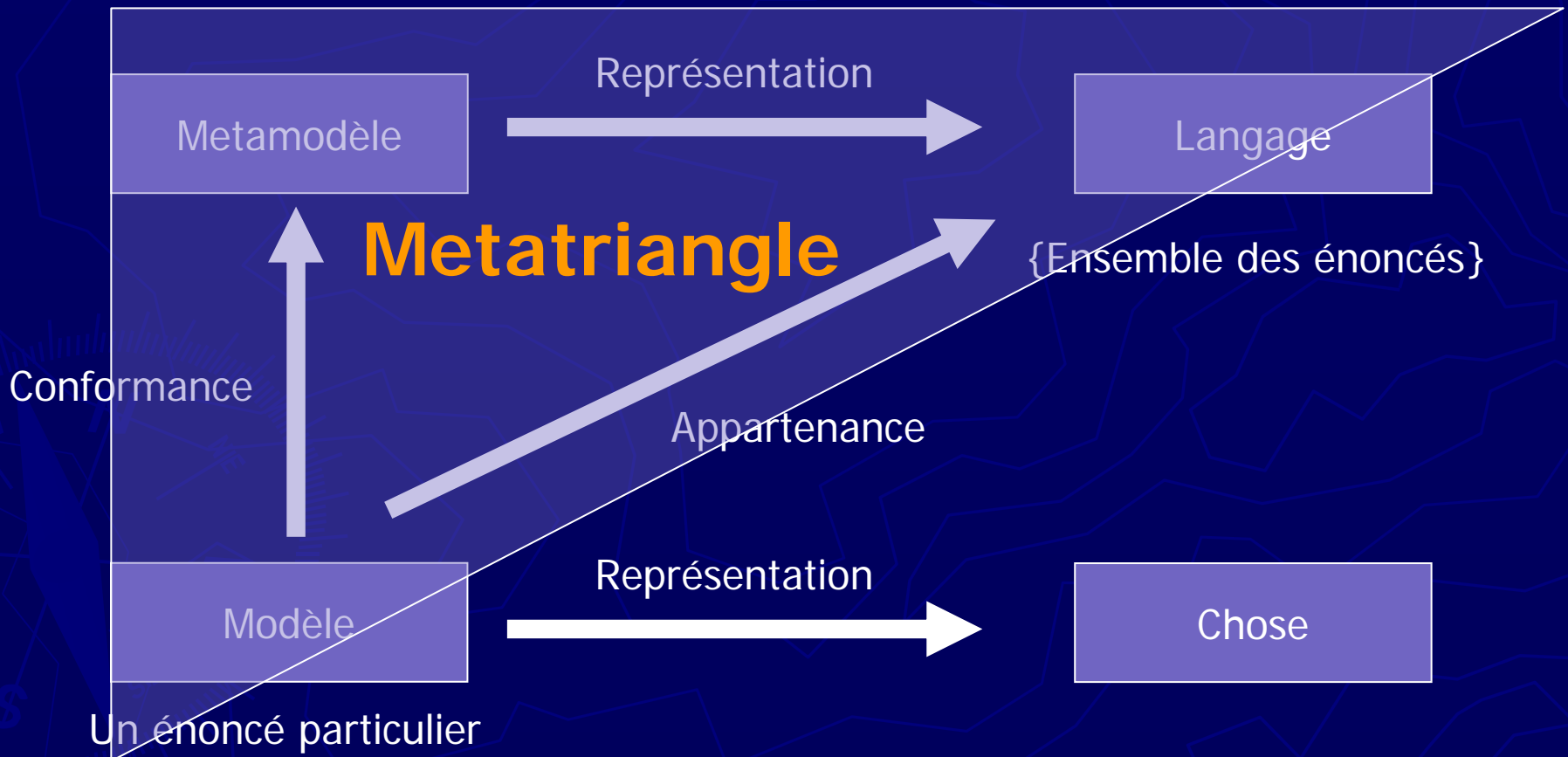
Le langage

- ▶ Les mots pour dire les choses
 - Outil pour la manifestation externe de la pensée
 - Plus ou moins généraliste



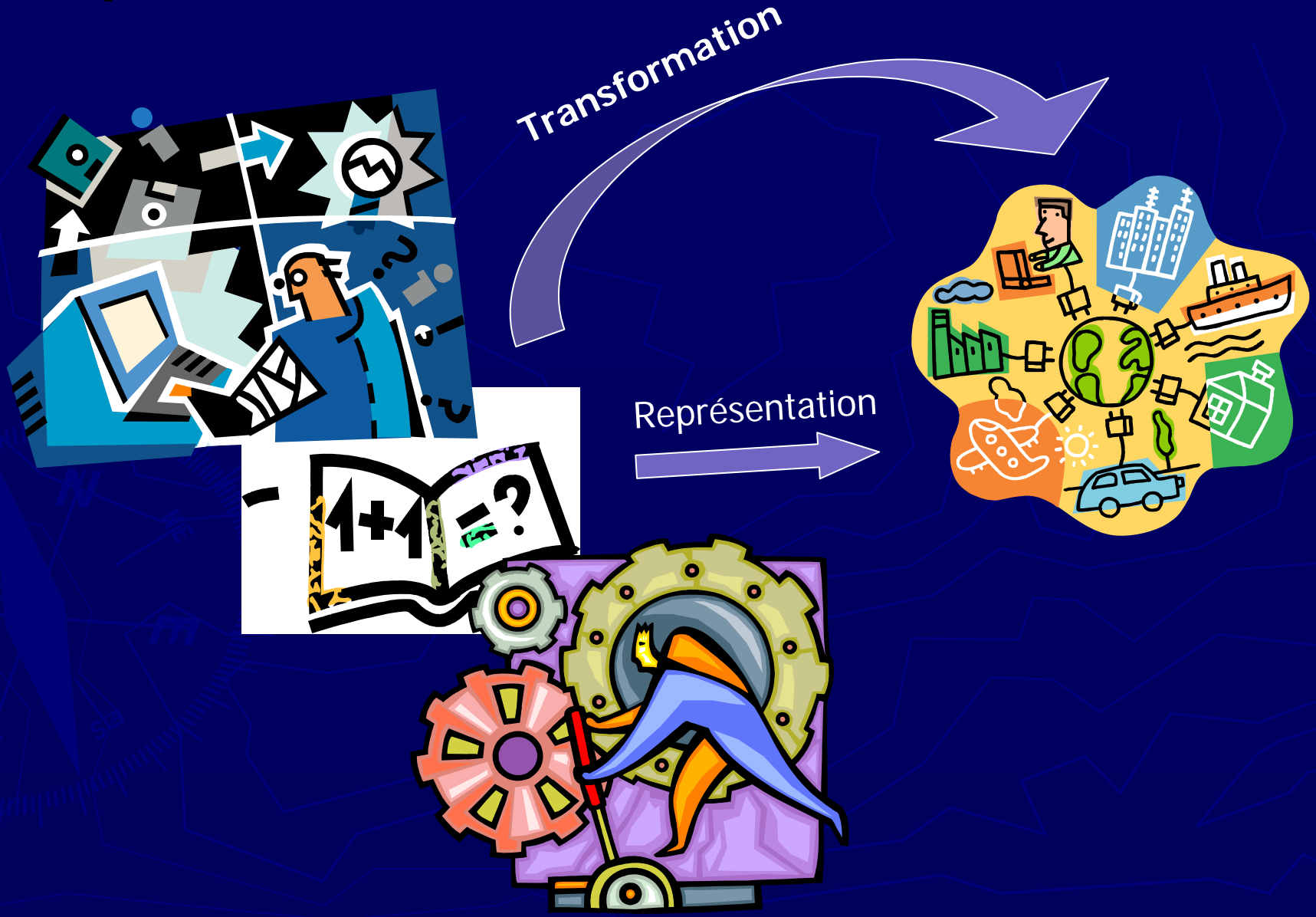


Relations modèles/metamodèles/langages





Opérationnalisation des modèles

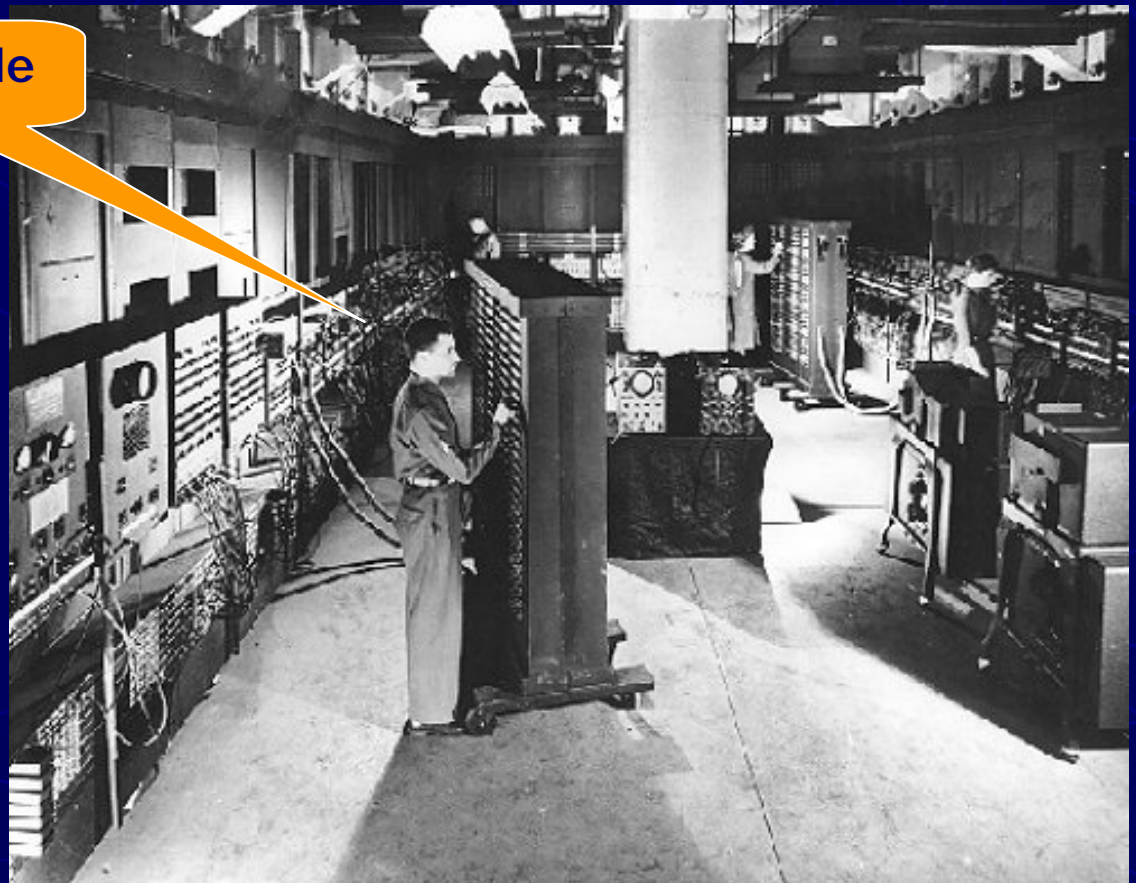




Arrivée de la machine

- ▶ Premiers pas vers l'opérationnalisation

Le modèle

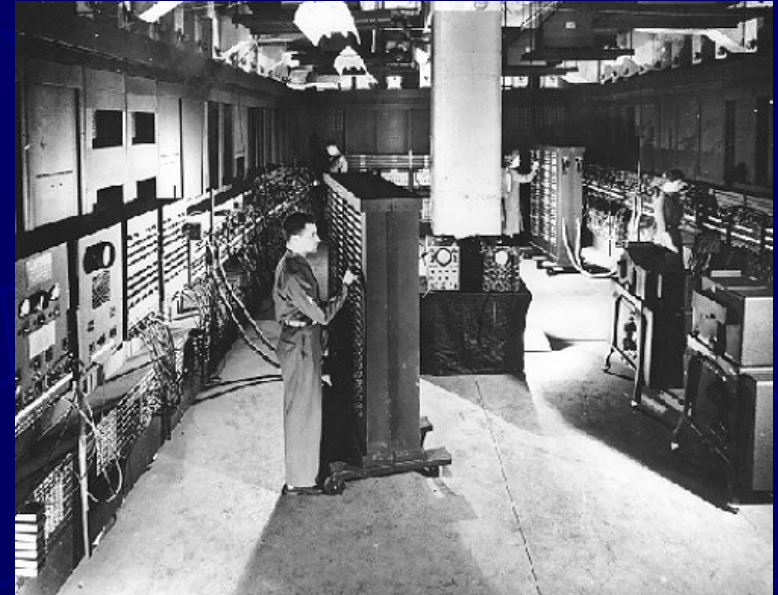




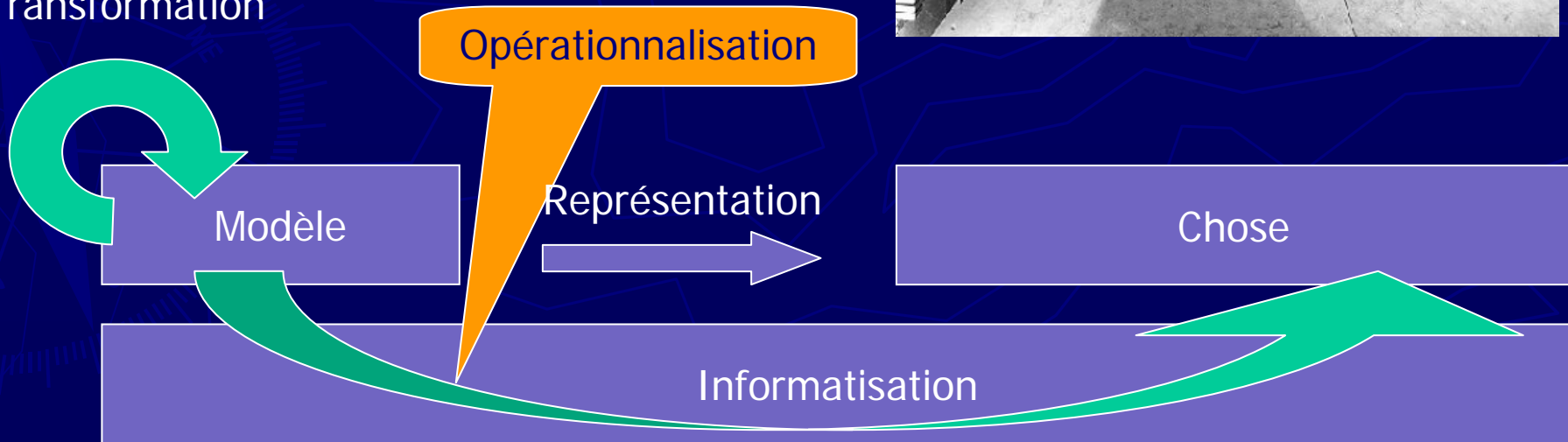
L'ordinateur : une chose « spéciale »

► Support

- des choses (en partie)
- des modèles
- des transformations

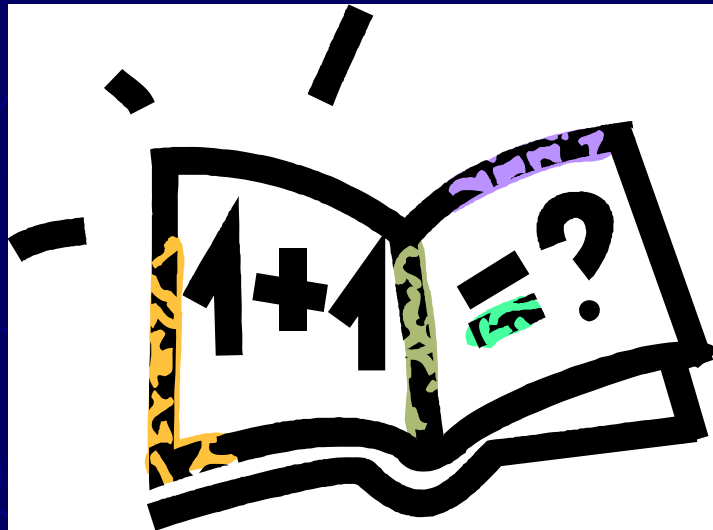


Transformation

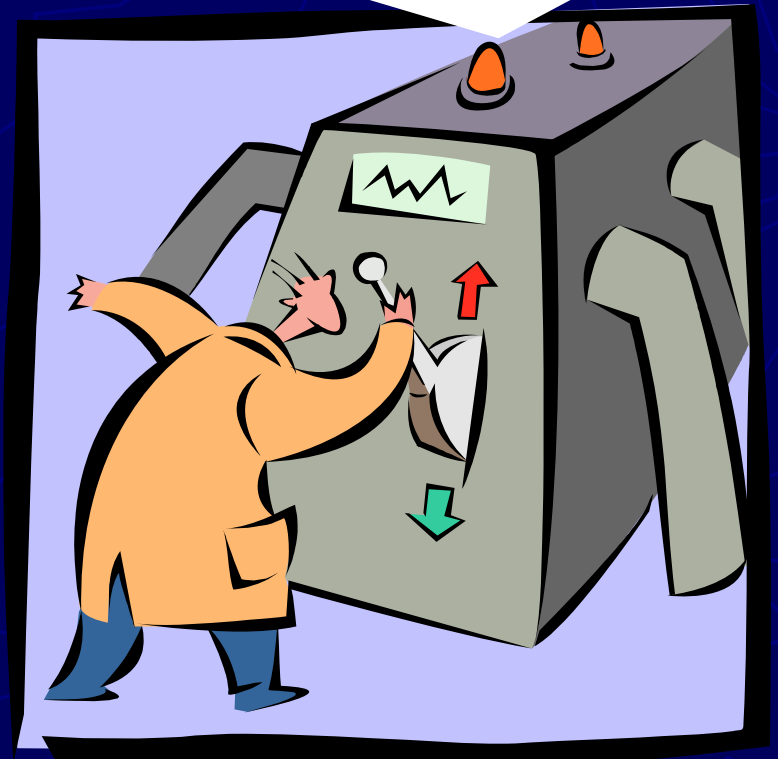




Une machine générique

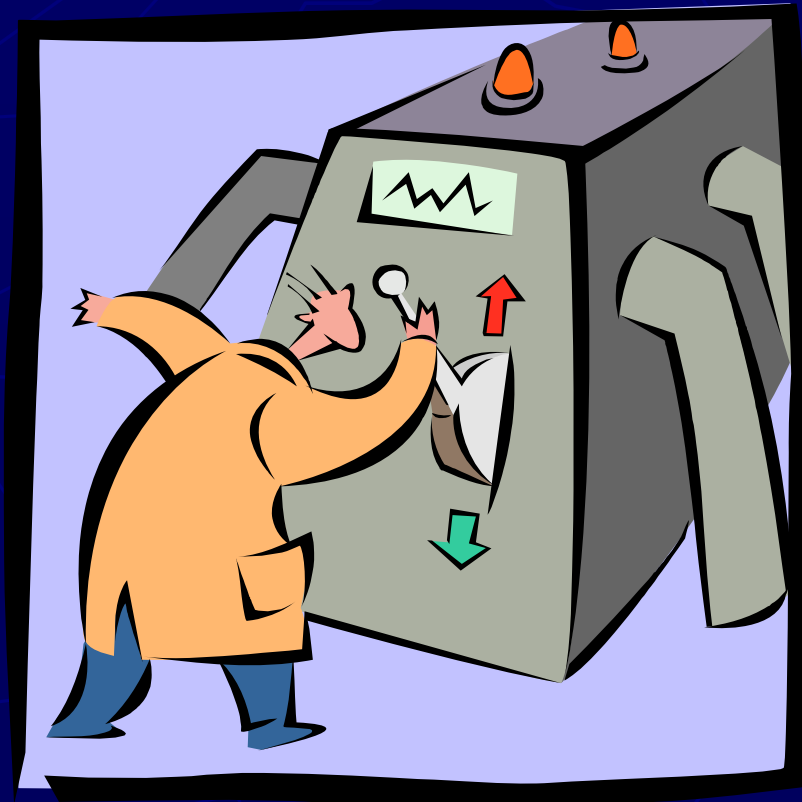


Paramétrisation





Et la machine devient le système

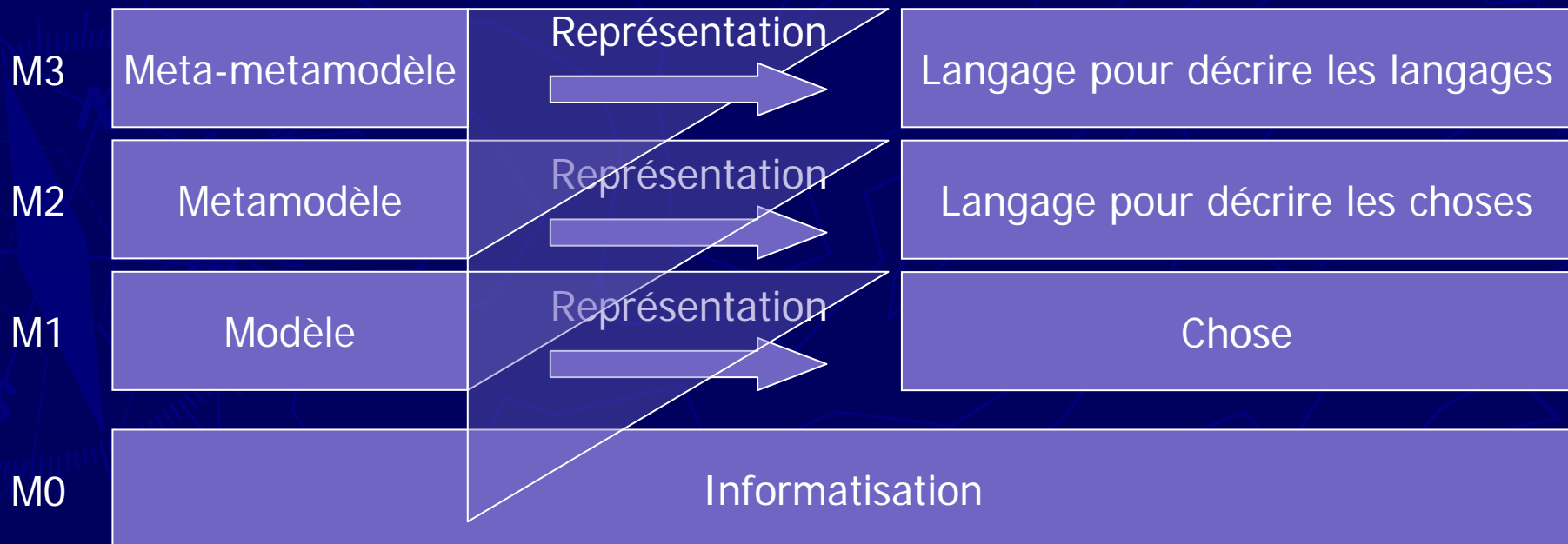




Architecture de metamodelisation

► MDA (Model-Driven Architecture)

- 2000
- www.omg.org/mda





Technologies du moment

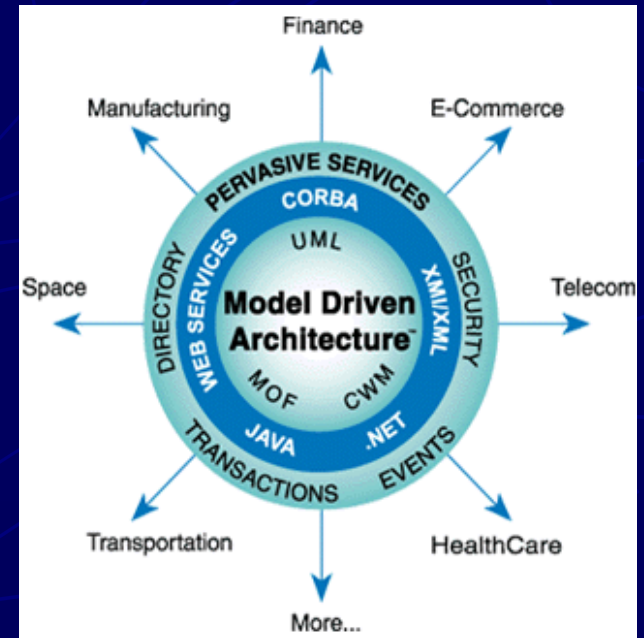


► Sphère MDA

- Normalisation MOF et UML

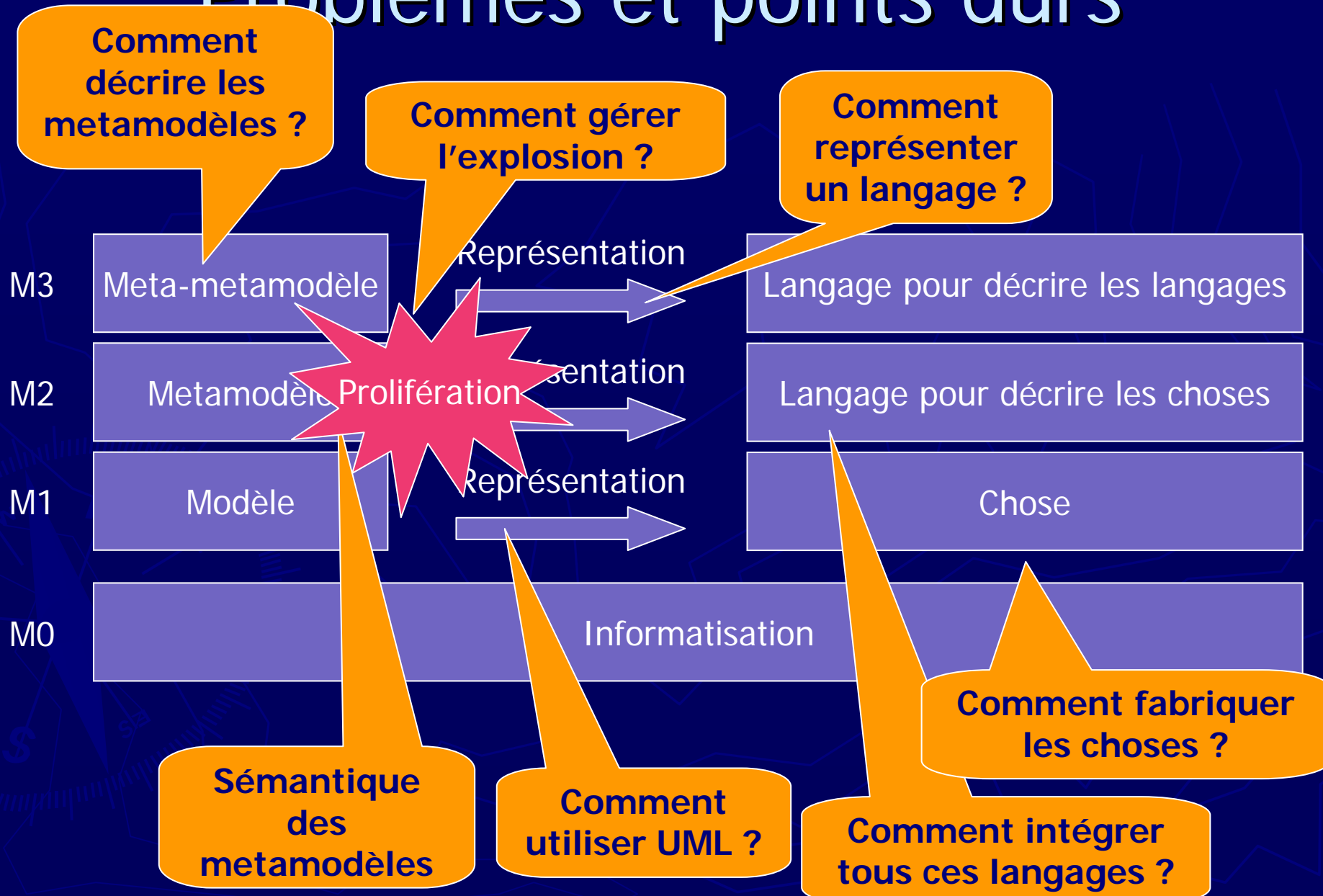
► Sphère Eclipse

- EMF/Ecore + Java





Problèmes et points durs





Metamodélisation exécutable

Système de metamodélisation

Syntaxes abstraite, et concrète, et sémantique

Comment décrire les metamodèle ?

Comment gérer l'explosion ?

Comment représenter un langage ?



Prolifération

Metamodèles métier

Système de metamodélisation

Opérationnalisation

Extrémités d'association

Démarche pour le déploiement d'UML

Comment fabriquer les choses ?

Sémantique des metamodèles

Comment utiliser UML ?

Comment intégrer tous ces langages ?



Milieu des années 90

► Nombreuses approches de modélisation par objets

- Booch 91-93
- OMT 90
- OOSE
- Classes-relations
- Hood
- ...



Contributions sur la
sémantique des
relations



1997 arrivée d'UML



▶ Normalisation par consensus

- Définition imprécise

Nombreux points
d'interprétation de
la sémantique

▶ Une notation

- Rien sur la démarche

Une boîte à outil
sans mode d'emploi



Contributions méthodologiques

▶ Démarche d'utilisation d'UML

- Pilotée par les cas d'utilisation
- Itérative, incrémentale
- Centrée sur l'architecture

Mise au point avec
Rational Software

Consolidée dans
l'ouvrage
*Modélisation objet avec
UML*

▶ Réutilisation de savoir-faire métier

- Métier de l'impression
- Métier de l'automaticien

Thèse CIFRE
Marie-Christine Roch

Thèse MNRT
Nathalie Gaertner



Publication en annexe

- ▶ J. Bezivin et P.-A. Muller. "UML: The Birth and Rise of a Standard Modeling Notation." In *Proceedings of the UML 1998 - ACM/IEEE 1st International Conference on The Unified Modeling Language*, pp. 1-8, Mulhouse, Lecture Notes in Computer Science, 1618, June 1998.



Metamodélisation exécutable

Système de metamodélisation

Syntaxes abstraite, et concrète, et sémantique

Comment décrire les metamodèle ?

Comment gérer l'explosion ?

Comment représenter un langage ?



Prolifération

Metamodèles métier

Système de metamodélisation

Opérationnalisation

Extrémités d'association

Démarche pour le déploiement d'UML

Comment fabriquer les choses ?

Sémantique des metamodèles

Comment utiliser UML ?

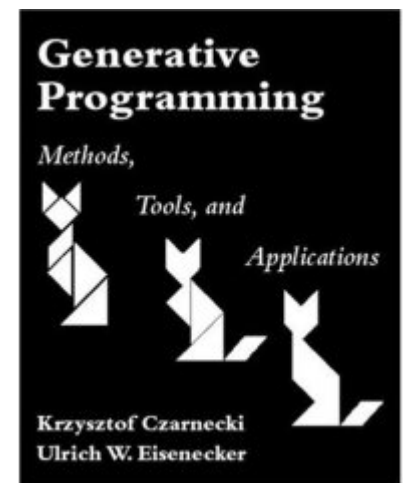
Comment intégrer tous ces langages ?



Fin des années 90

- ▶ **Opérationnalisation envisageable**
 - Notion de lignes de produits
 - Programmation générative
 - Meilleure définition d'UML
- ▶ **Validation industrielle**
 - **Priorité**

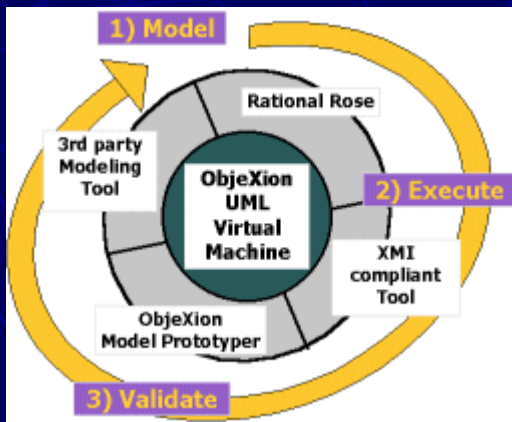
L'opérationnalisation des modèles est-elle viable dans un contexte industriel ?





Transfert de technologie

- ▶ Création d'ObjeXion Software SA (1999)
 - Outils de prototypage de modèles (UML)
 - Modélisation opérationnelles de systèmes d'information Web
 - ▶ Première solution IDM pour le Web





Pourquoi du GL pour le web ?

▶ Étude du Cutter Group (2000)

- 84% des projets web ne répondent pas aux besoins métiers exprimés
- 79% des projets web ne maîtrisent pas les délais
- 63% des projets web dépassent le budget
- 53% des projets web livrés ne fournissent pas les fonctionnalités demandées
- La documentation est inexistante ou illisible dans 53 % des projets



Premiers pas : Model-Prototyper

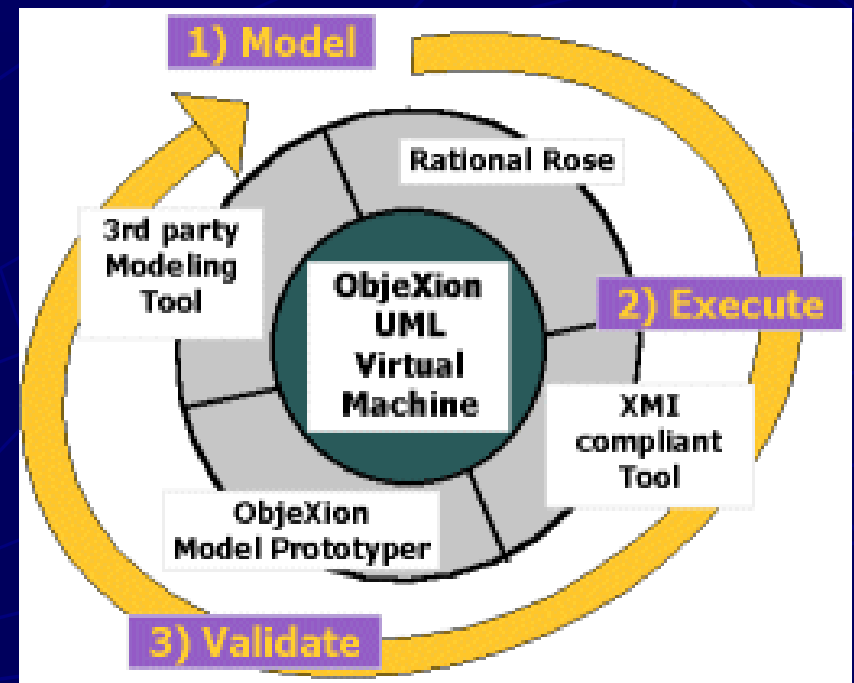
► Problème

- Les utilisateurs d'UML ne réalisent pas la signification de leurs modèles

► Solution traditionnelle

- Actions de conseil
- Audit de modèles

Problème, cette approche ne passe pas à l'échelle





Prototypage de modèle

Problème, la valeur perçue des prototypes est faible. Il faut réaliser de vraies applications

Exécution de scénarios

Analyse du modèle et génération d'un prototype

For Help, press F1

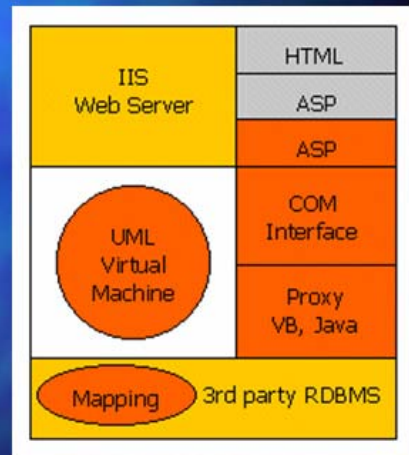


Aller au-delà des prototypes

Web servers integration

Objexion
Software

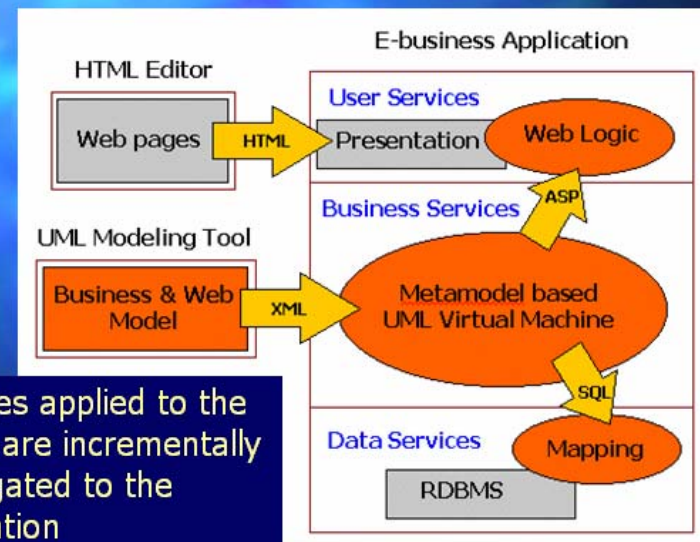
- Metamodel level
 - COM interface
 - Wizards
- Model level
 - Language proxies
 - Method definition



Components in orange get incrementally updated

The overall picture

Objexion
Software

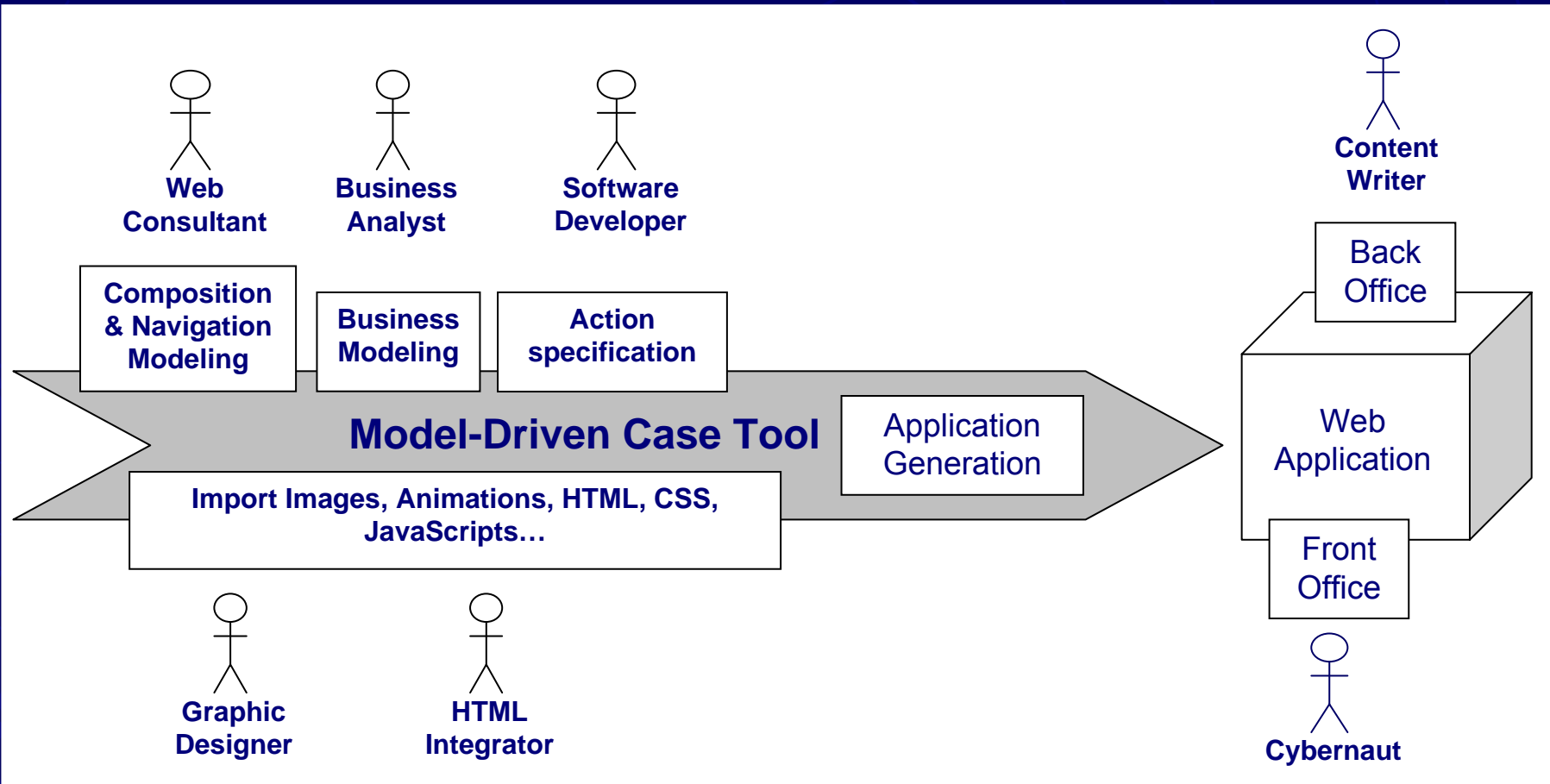


Changes applied to the model are incrementally propagated to the application

**Problème, cette
approche ne passe
pas à l'échelle**

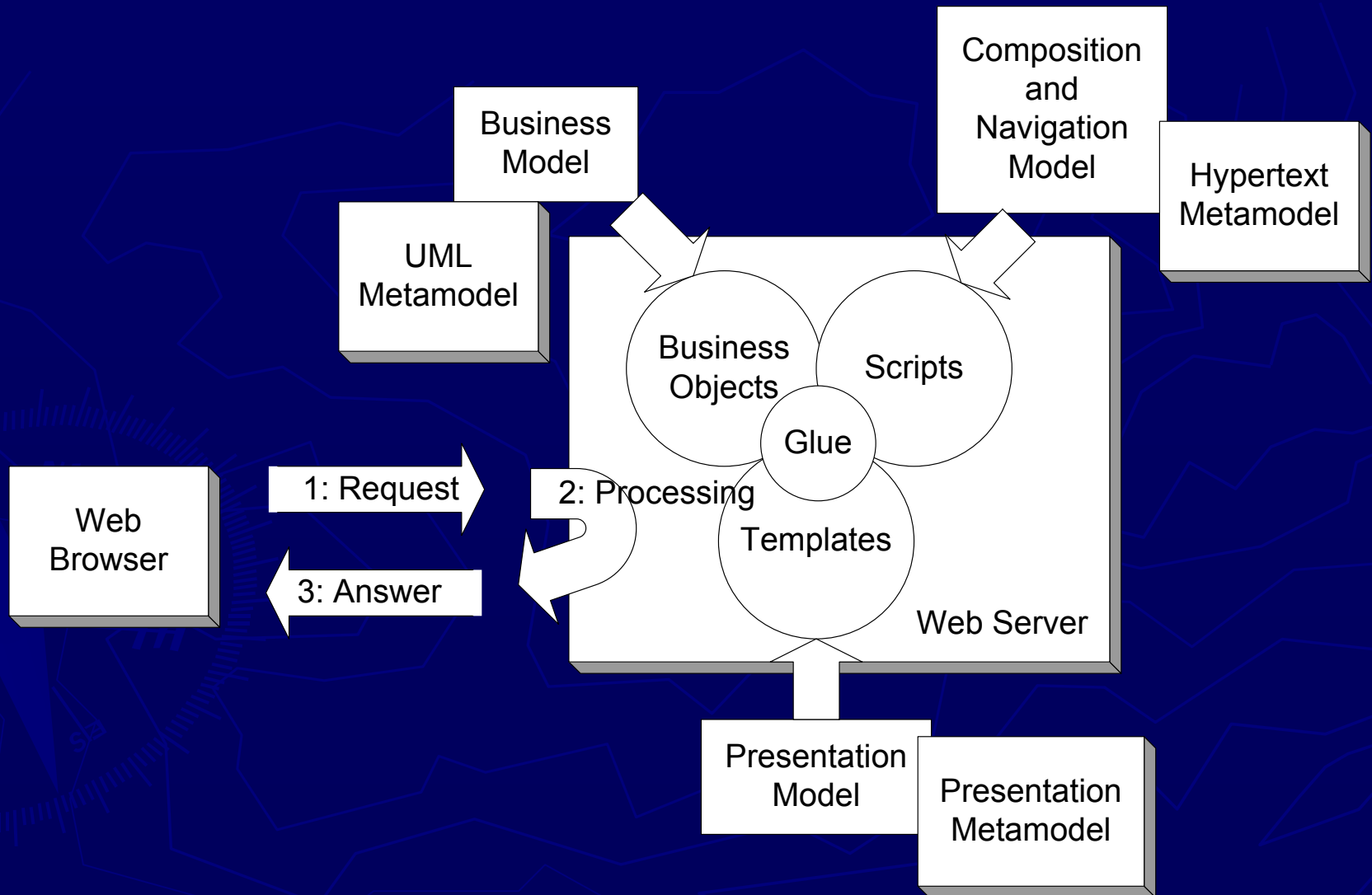


Passage à l'échelle



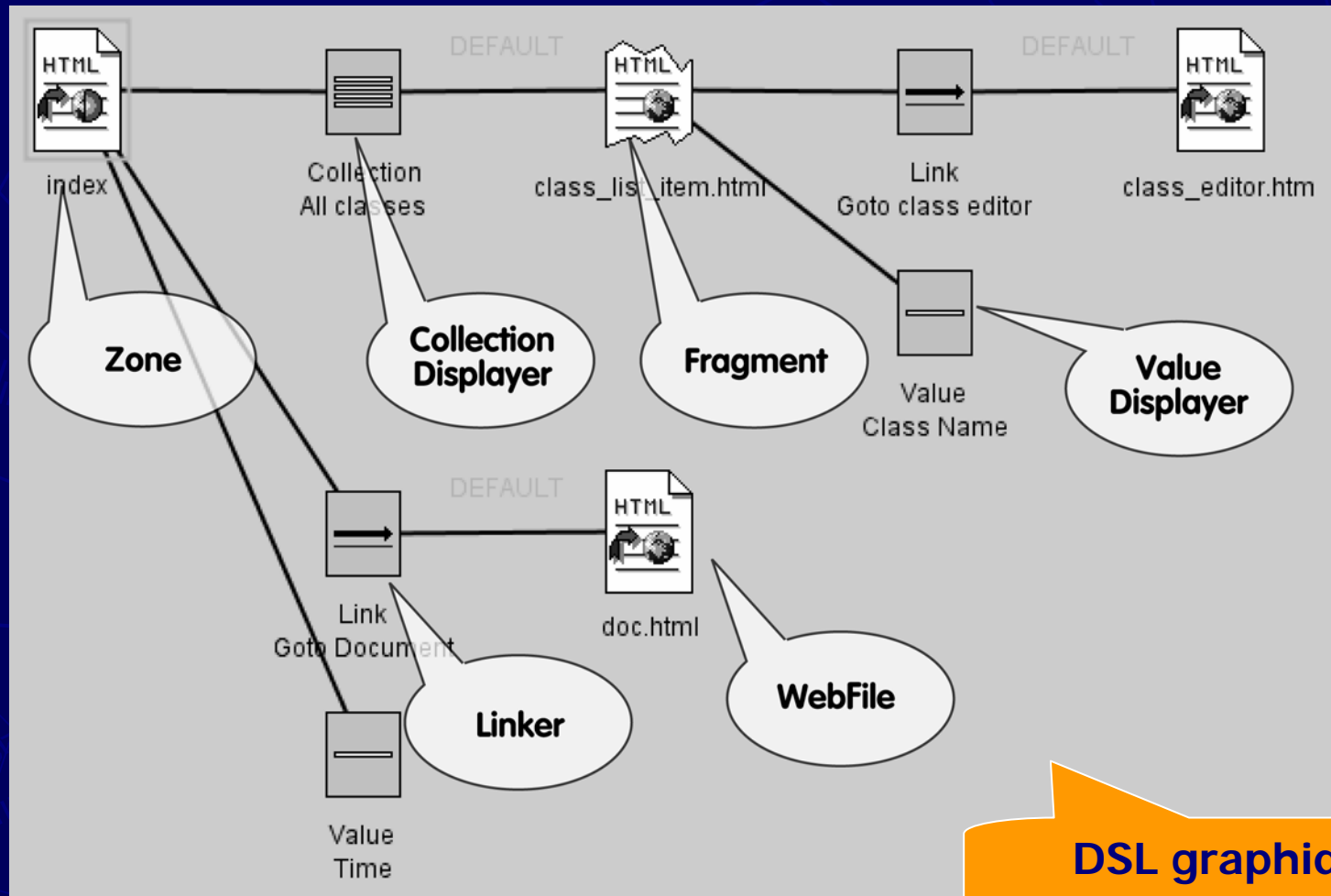


Modélisation multi-modèles





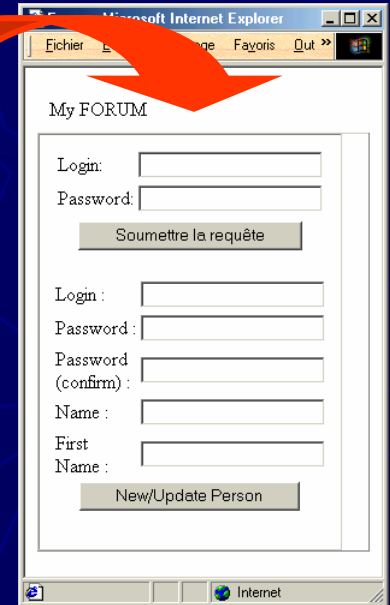
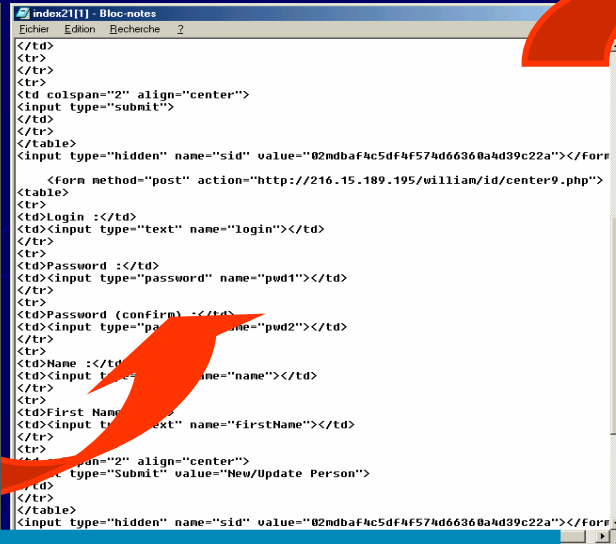
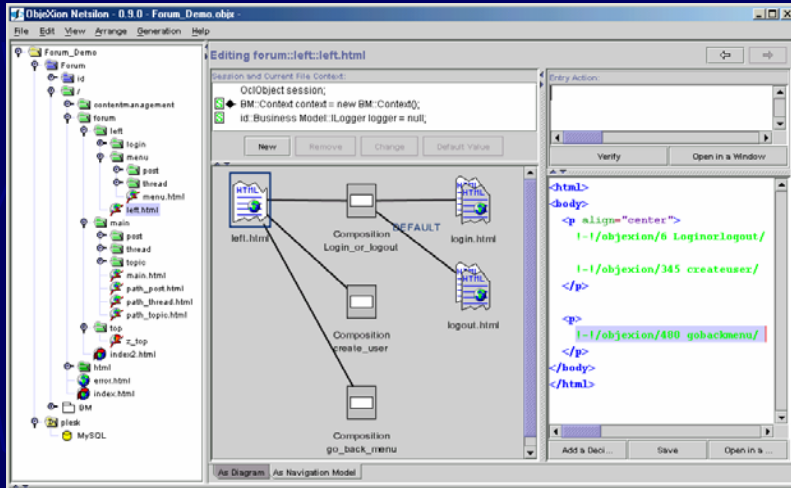
Modèle de navigation



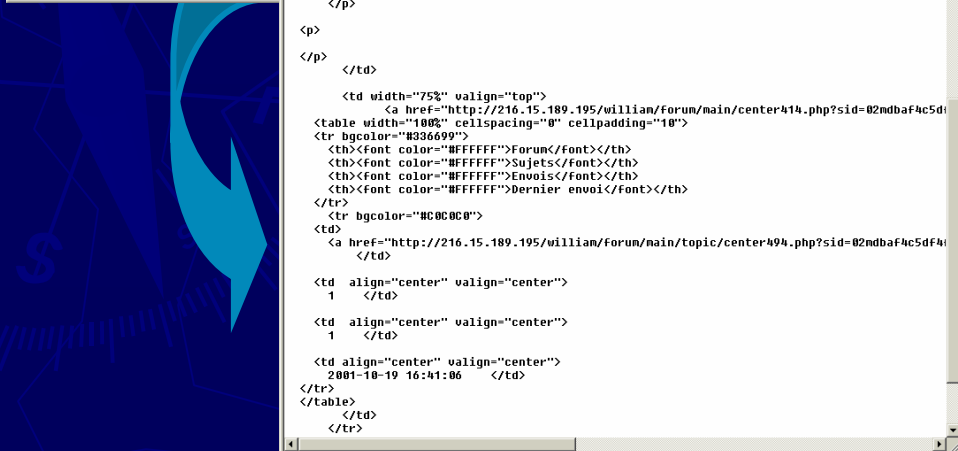
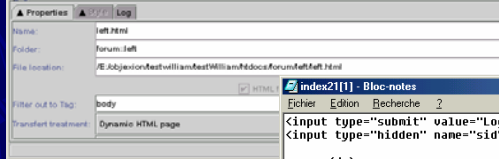
DSL graphique



Netsilon en action



Facteur d'expansion de 20

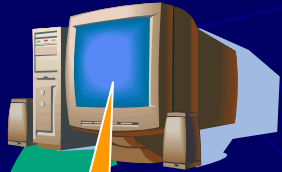


**Problème,
Facteur insuffisant,
il faut viser deux ordres
de grandeurs.
(wizzards)**



PIM / PSM

Netsilon



Le PSM
contraint le
PIM

ServeurS
HTTP
Apache
IIS
Netscape

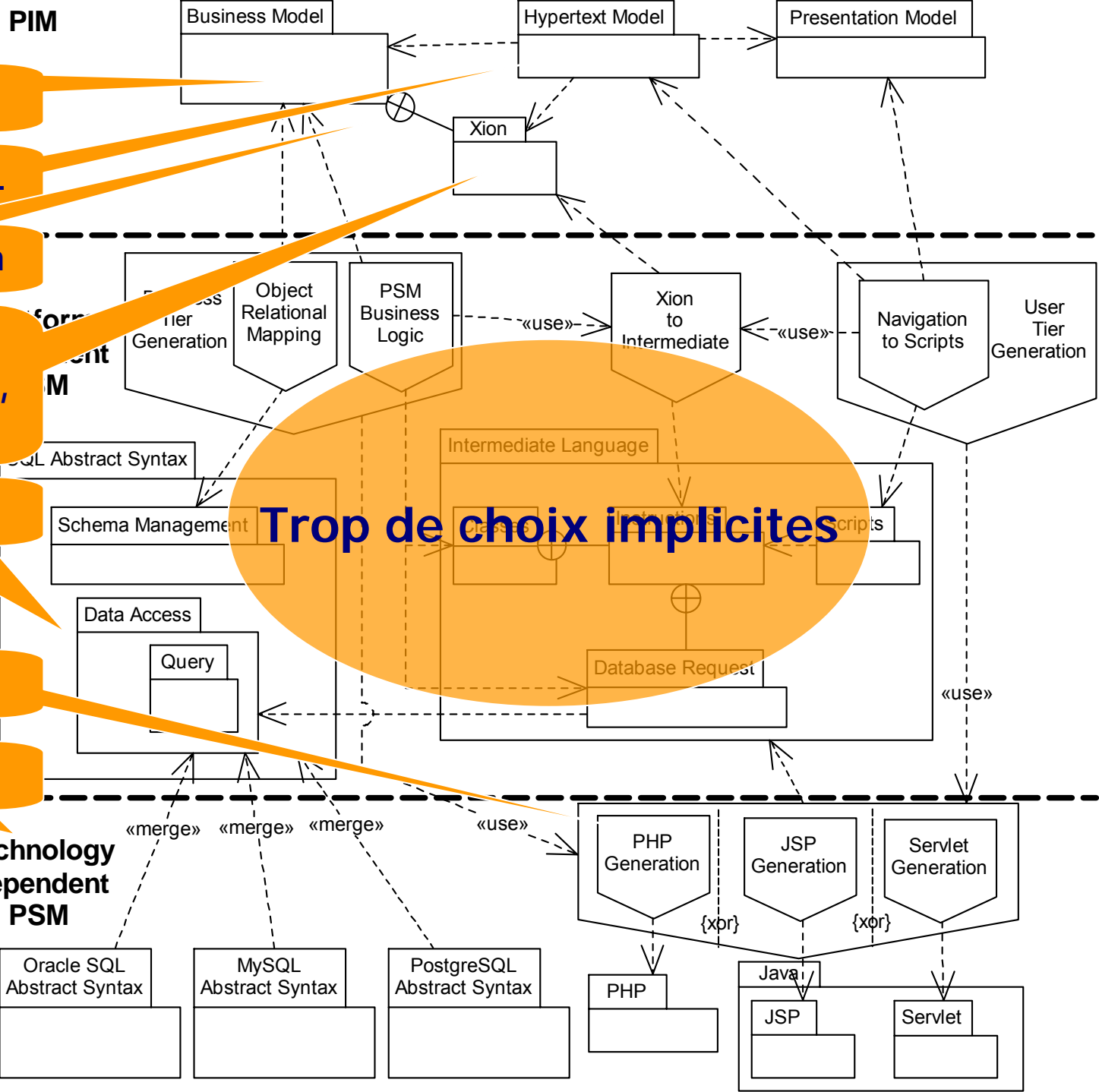
ServeurS
J2EE
Tomcat
Resin
Weblogic
Enhydra

ServeurS
PHP
PHP Module
APC (Cache)
Zend (Cache)

Problème,
L'indépendance
technique
n'intéresse pas
les praticiens !



ServeurS
Données
Oracle
MySQL



Pas UML

Nouveau DSL

Pas de fusion

Manque introspection, et exception

Patrimoine ?

Image ?

Intérêt ?

Trop de choix implicites



Publication en annexe

- ▶ P.-A. Muller, P. Studer, F. Fondement, et J. Bézivin. "Platform Independent Web Application Modeling and Development with Netsilon", *Software and Systems Modeling*, vol. 4 (4), pp. 424-442, 2005.



Rendre plus explicite

The TopModL Initiative

www.topmodl.org

An
Open-Source
Platform
for
Research
on
Model-Driven
Engineering



Sébastien GÉRIARD



Jérôme DELATOUR



Frédéric FONDEMENT



Jean-Marc JEZEQUEL



Cédric DUMOULIN



Pierre-Alain MULLER

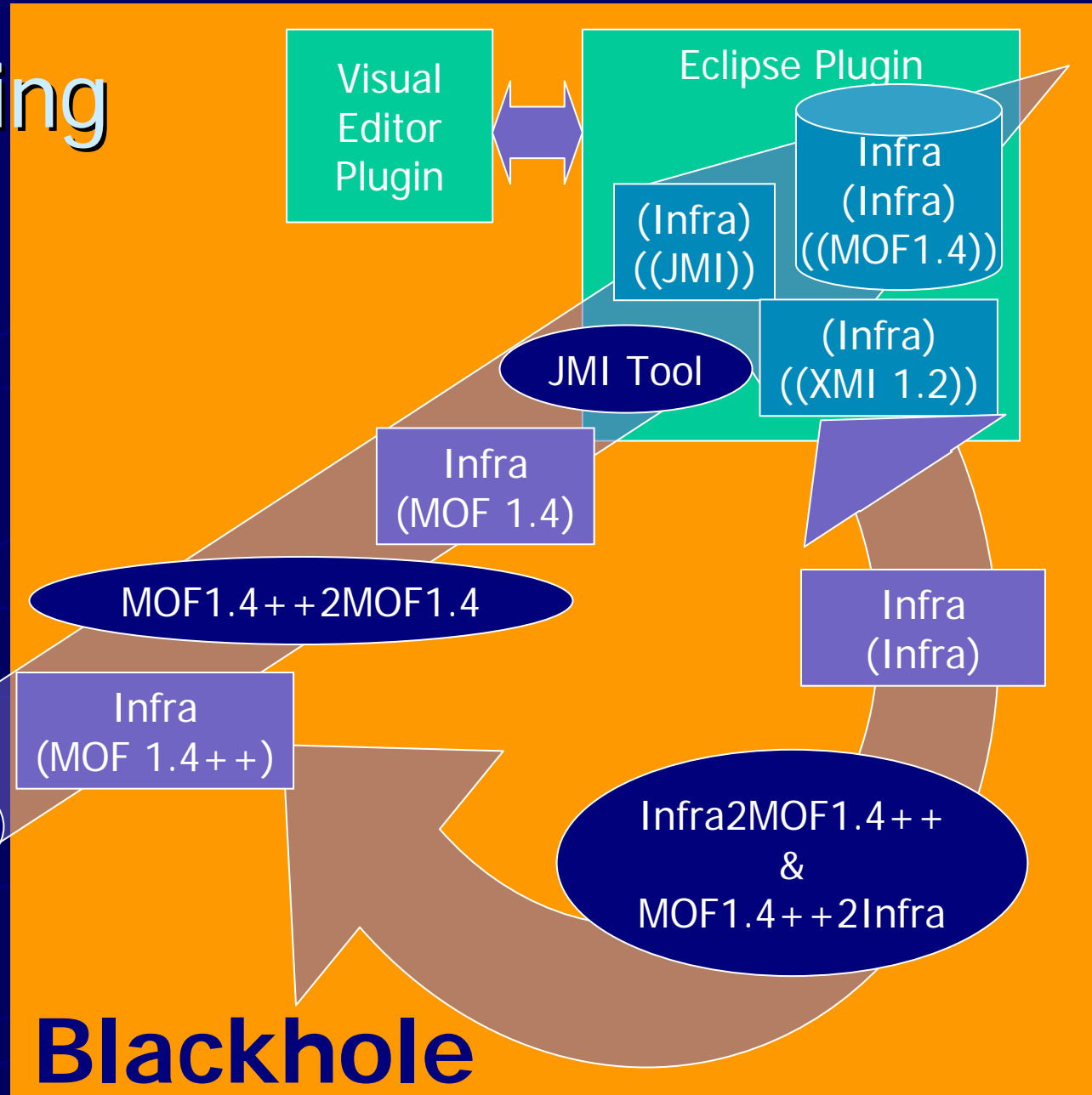
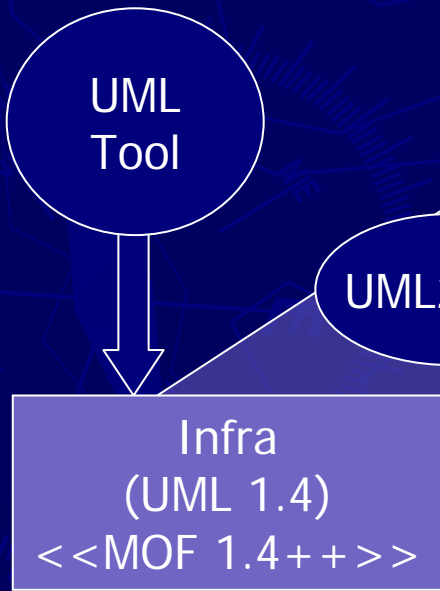


Principes retenus

- ▶ Le fait que tout est modèle,
- ▶ Les notions de langages, modèles et rôles,
- ▶ Le fait que TopModL lui-même est un modèle,
- ▶ Le fait que tout est explicite, y compris l'architecture de metamodelisation,
- ▶ L'indépendance par rapport aux référentiels.



Bootstrapping TopModL





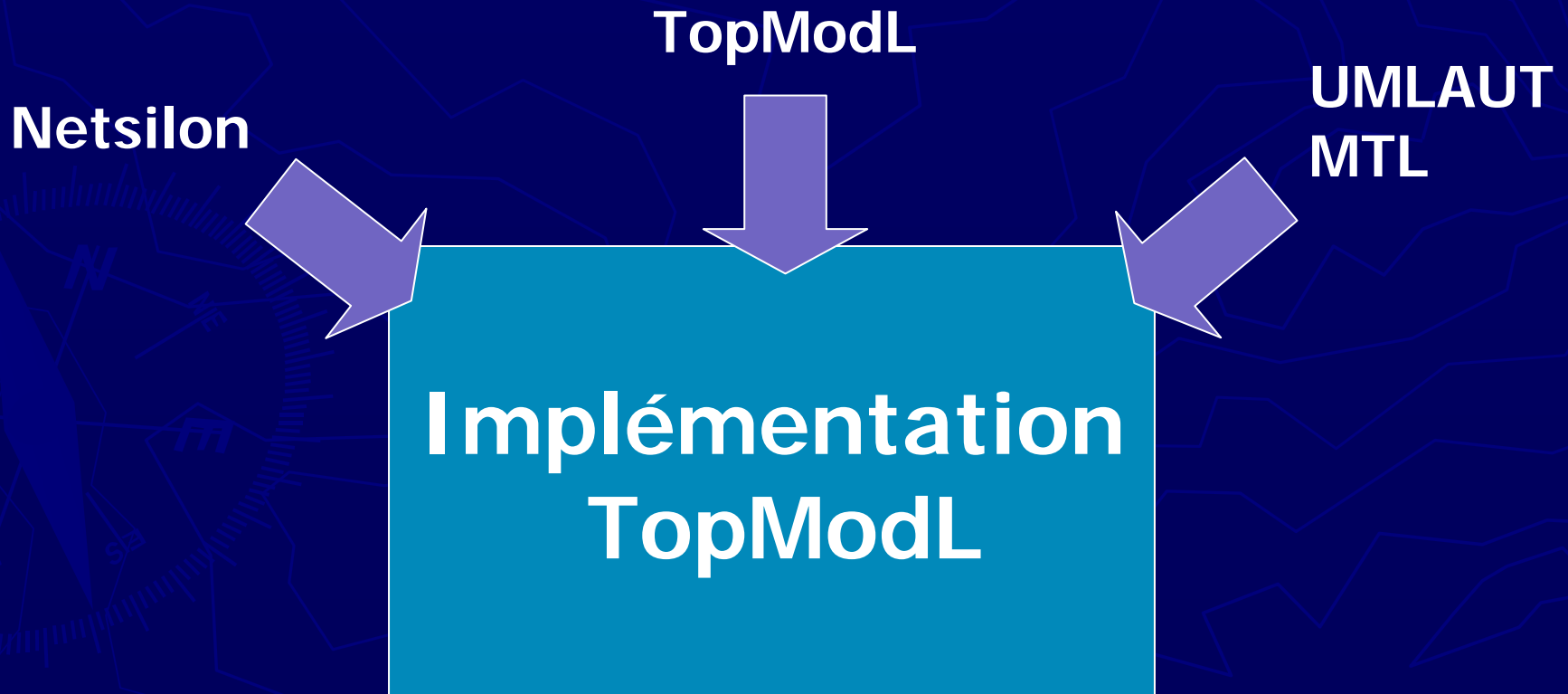
Principaux enseignements

- ▶ Les avantages de se rendre indépendant du niveau M3 (tel que définit par l'OMG) ne sont pas contrebalancés par les inconvénients de créer un nouveau langage M3 pivot.
- ▶ Il est plus judicieux de rechercher la compatibilité maximale avec un langage de metadonnées existant qui dispose déjà d'un grand nombre d'outils.
- ▶ Il est difficile de trouver une source de financement pour un projet sur la metamodélisation en générale, il faut rechercher un domaine d'application (ce que je ferai dans le cadre du montage du projet RNTL OpenEmbeDD).



Implémenter la vision TopModL

- ▶ Délégation INRIA dans l'équipe Triskell





Kermeta





Metamodélisation exécutable

Système de metamodélisation

Syntaxes abstraite, et concrète, et sémantique

Comment décrire les metamodèle ?

Comment gérer l'explosion ?

Comment représenter un langage ?



Prolifération

Metamodèles métier

Système de metamodélisation

Opérationnalisation

Extrémités d'association

Démarche pour le déploiement d'UML

Comment fabriquer les choses ?

Sémantique des metamodèles

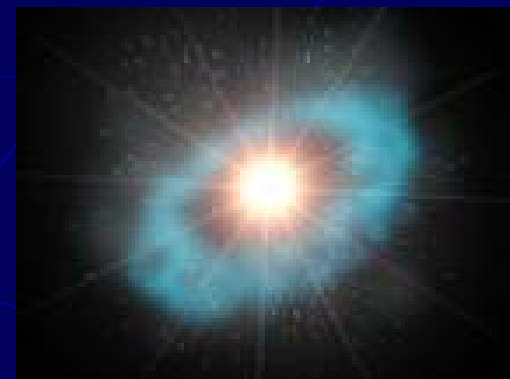
Comment utiliser UML ?

Comment intégrer tous ces langages ?



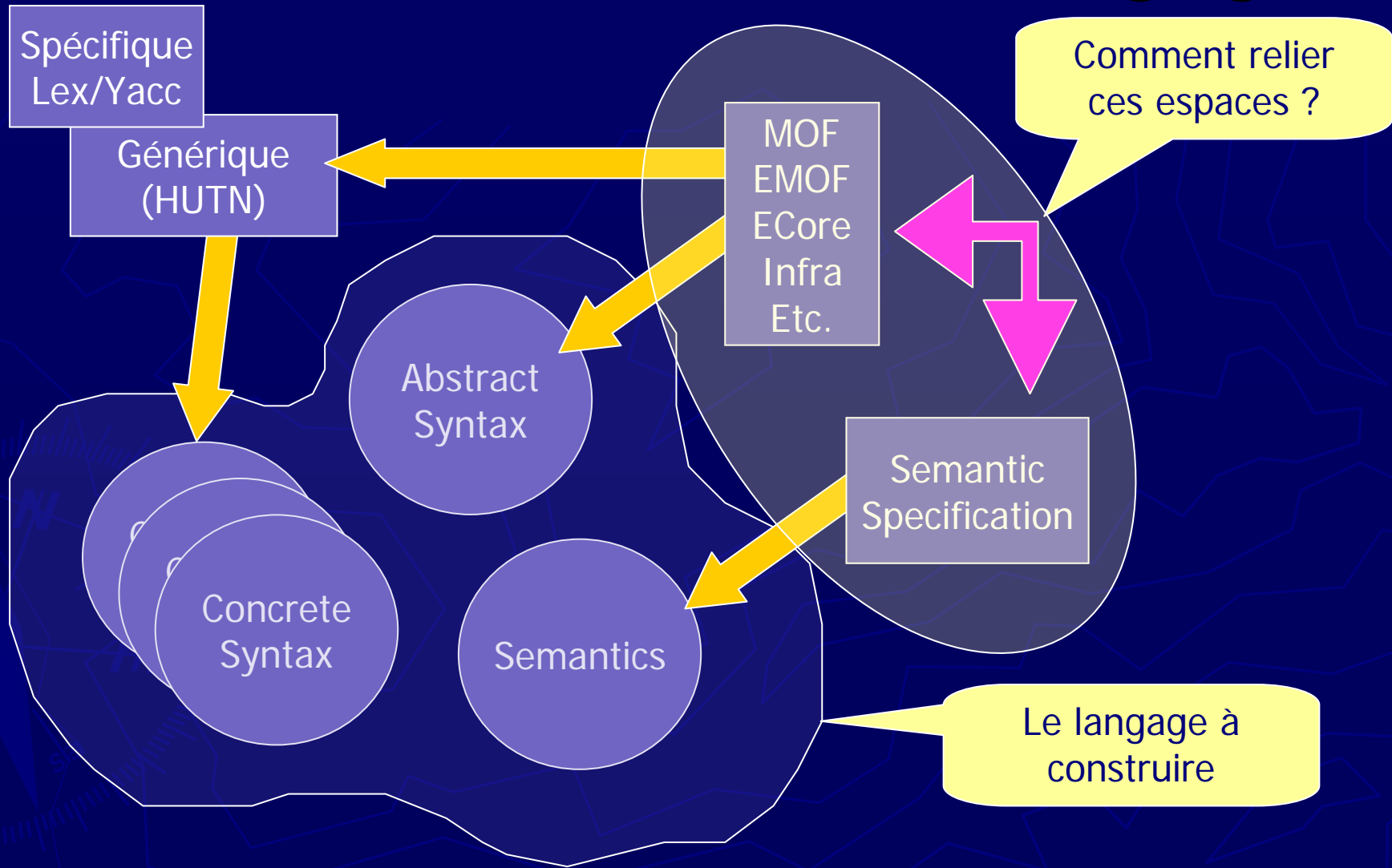
Prolifération

- ▶ De metamodèles
- ▶ De langages spécifiques à l'IDM
 - Transformations: QVT, ATL, MTL...
 - Contraintes: OCL
 - Actions: AS, Xion...
- ▶ De langages connexes
 - XSLT, XMI...





Des metamodèles aux langages





Langages de metadonnées

- ▶ (E)MOF => Aspects structurels
 - classes, propriétés, références...
 - opérations mais pas les méthodes
- ▶ Pas de support pour agir sur les modèles
 - Contraintes
 - Actions (CRUD sur des éléments)
 - Transformations
 - ...



Exemple typique (extrait de la spec du MOF)

- ▶ **Operation *isInstance(element : Element) : Boolean***
 - *"Returns true if the element is an instance of this type or a subclass of this type. Returns false if the element is null".*

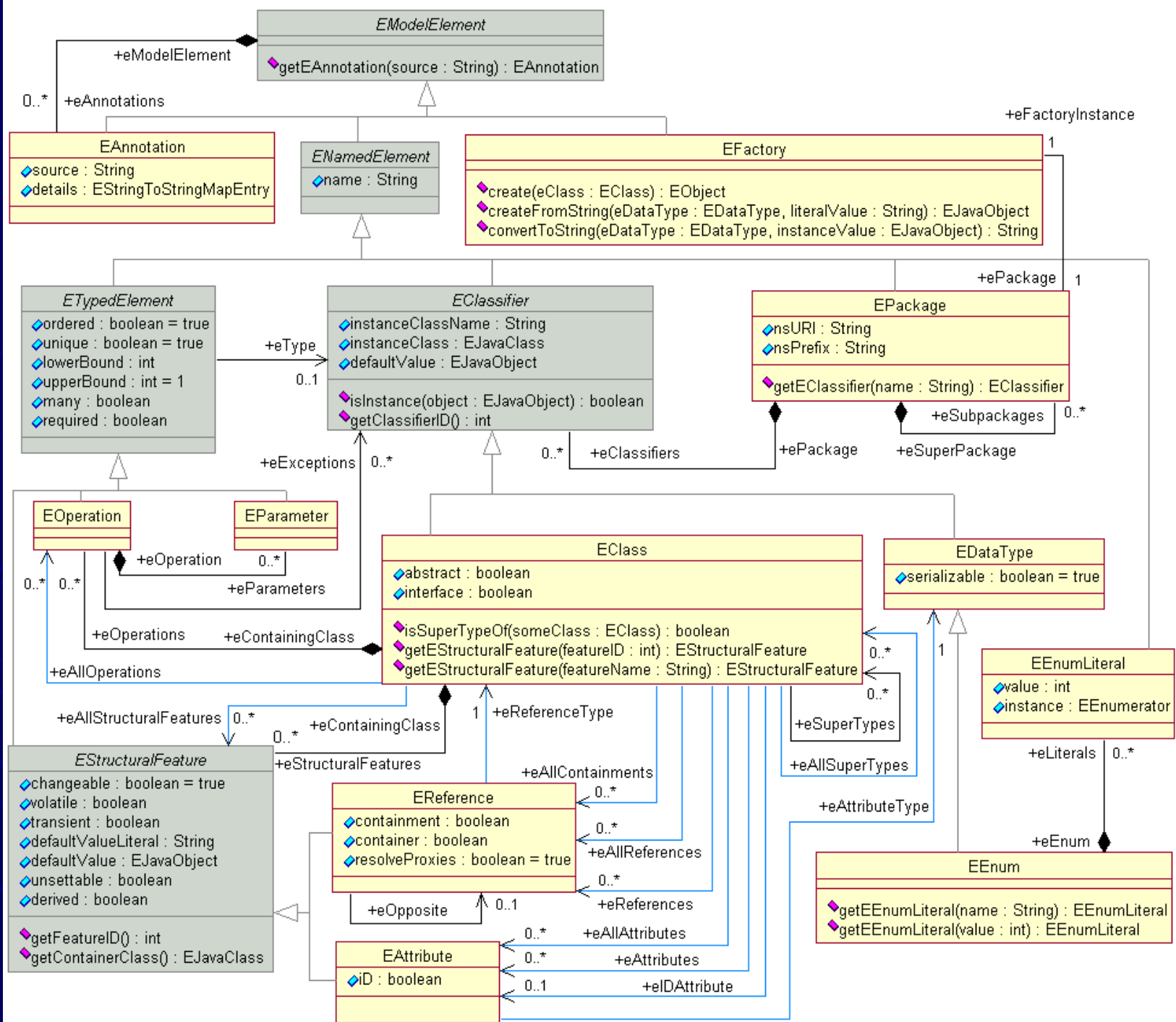
Spécification en langage naturel

```
operation isInstance (element : Element) : Boolean is do
  // false if the element is null
  if element == void then result := false
  else
    // true if the element is an instance of this type
    // or a subclass of this type
    result := element.getMetaClass == self or
              element.getMetaClass.allSuperClasses.contains(self)
  end
end
```

Spécification opérationnelle



Metamodeling Core





Problèmes

- ▶ Multiplicité de langages différents
 - Différentes approches, niveaux de détail, systèmes de types
 - Interopérabilité difficile
- ▶ Metamodèles structurels
 - Pas de comportement en natif
 - Projections vers d'autres langages



Vers la "meta"-exécutabilité ?

- Opérations de base CRUD
- Merge, Composition...



Definition



Execution

► Un programme qui manipule des éléments de modélisation

"Program = Data Structure + Algorithm", Niklaus Wirth



Meta-
Exécutabilité

=

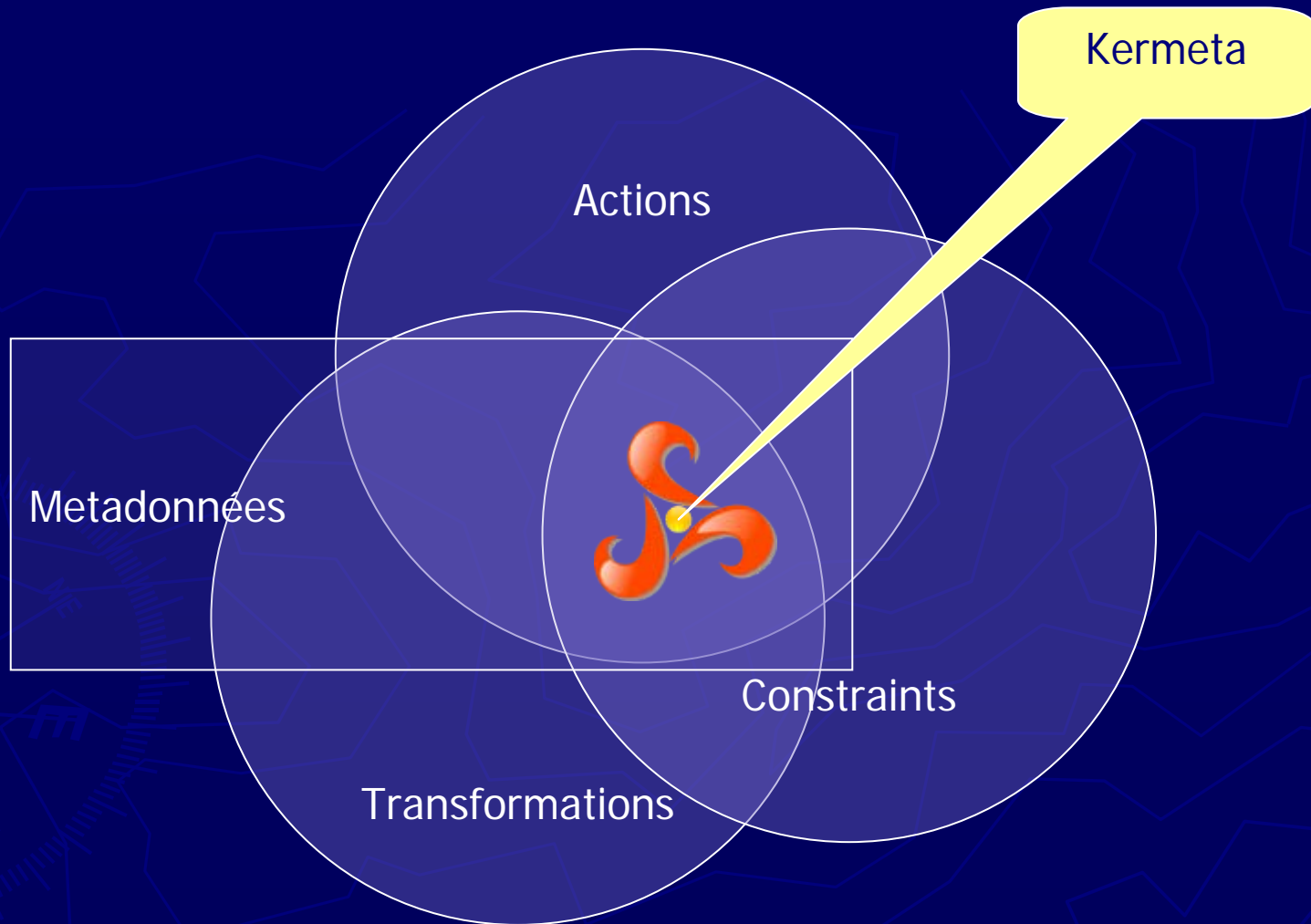
Metadonnées

+

Actions

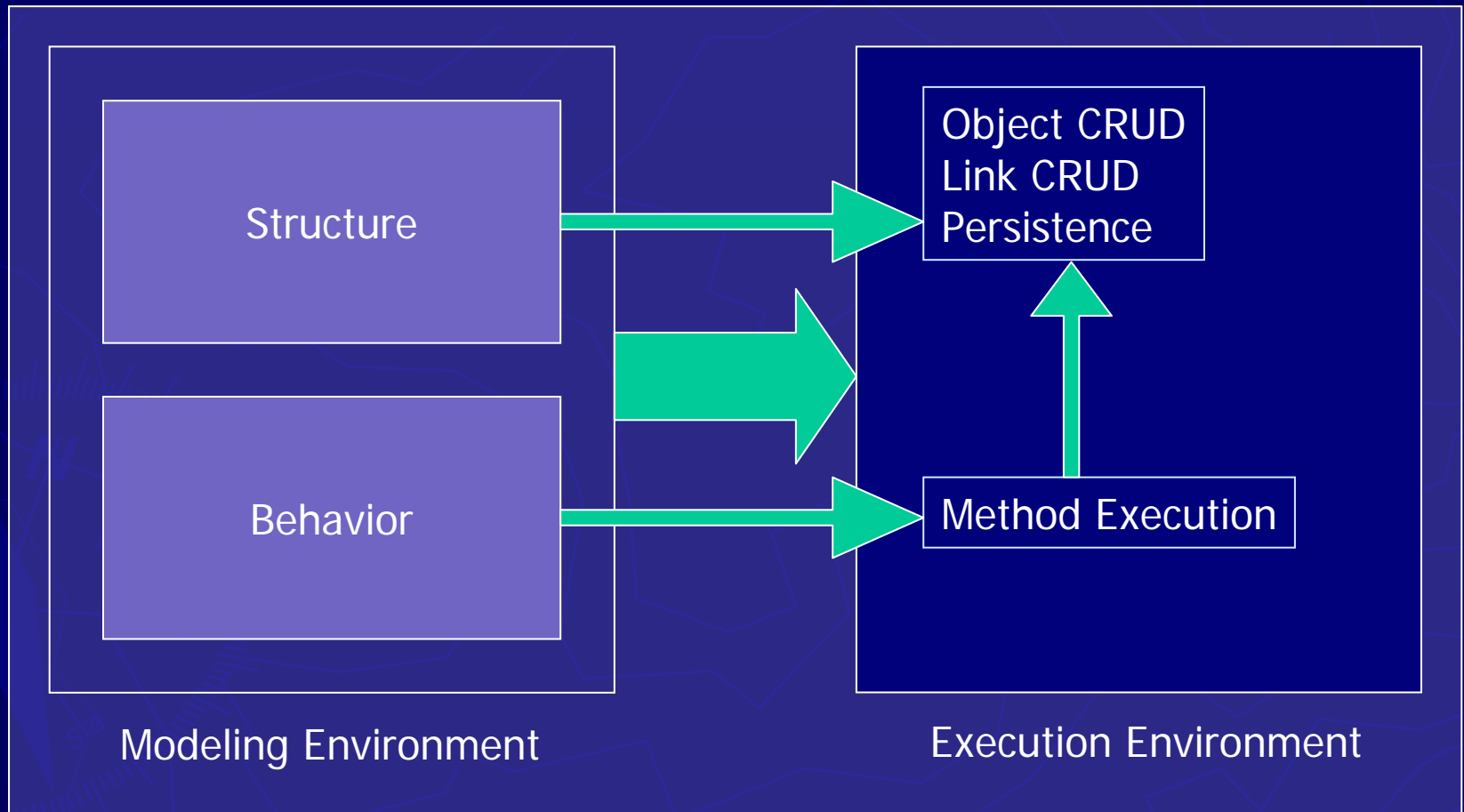


Kermeta, un noyau Meta





Environment d'exécution





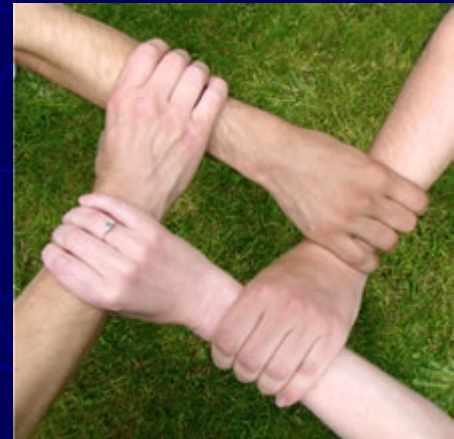
Comment construire Kermeta?

- ▶ Metadonnées + projections en Java
 - Solution de facilité, ou solution optimale ?
- ▶ Définir un nouveau langage
 - *Éventuellement un Pivot (TopModL)*
- ▶ Etendre un langage de metadonnées
 - Ecore++, EMOF++...



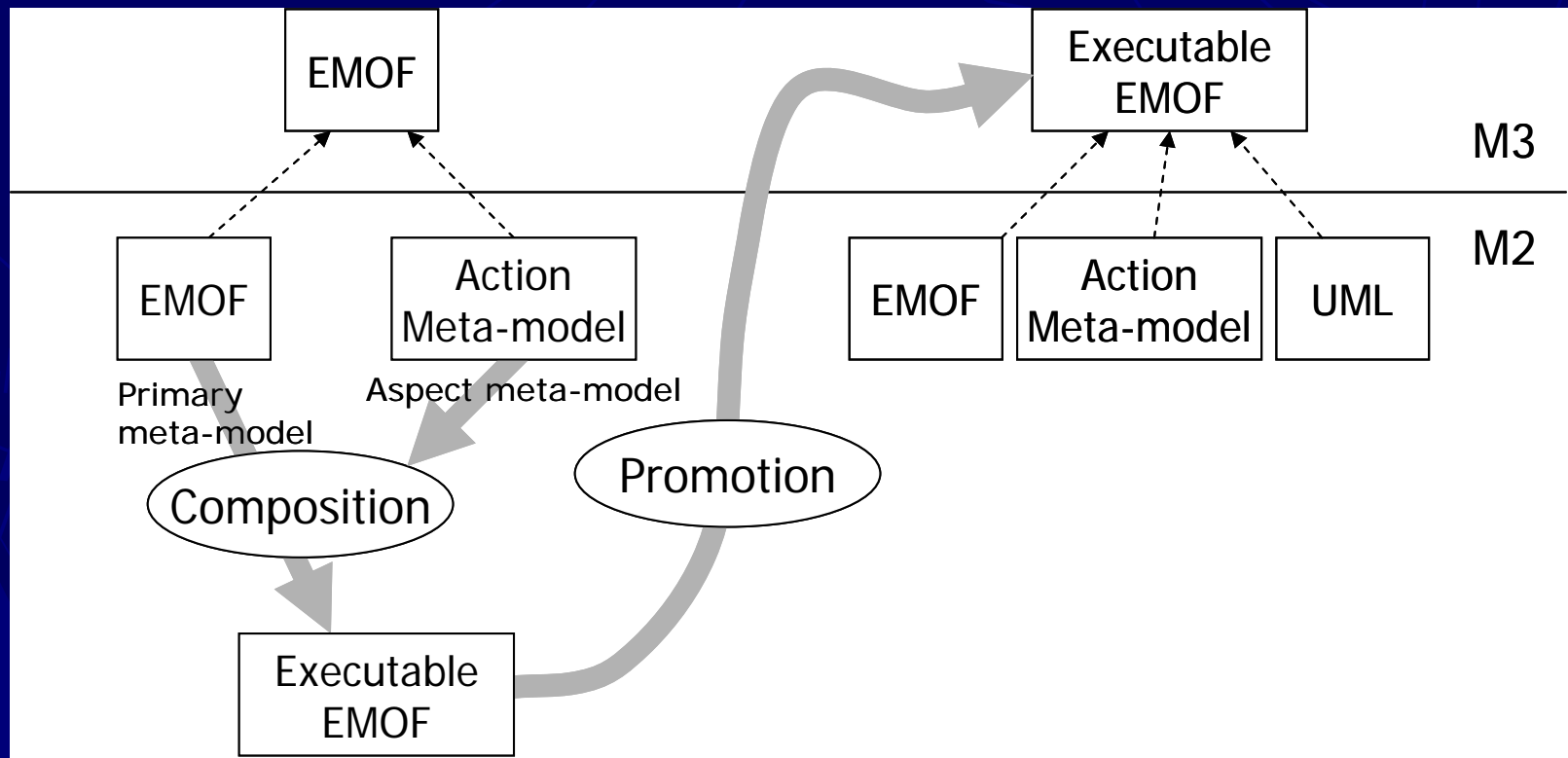
Choix pour Kermeta

- ▶ Compatible avec EMOF
- ▶ Orienté-objet
 - Héritage multiple
 - Rédefinition des méthodes et liaison dynamique
 - Réflexion (en lecture seule pour le moment)
- ▶ Typage statique
 - Généricité (classes et fonctions)
 - Type fonction



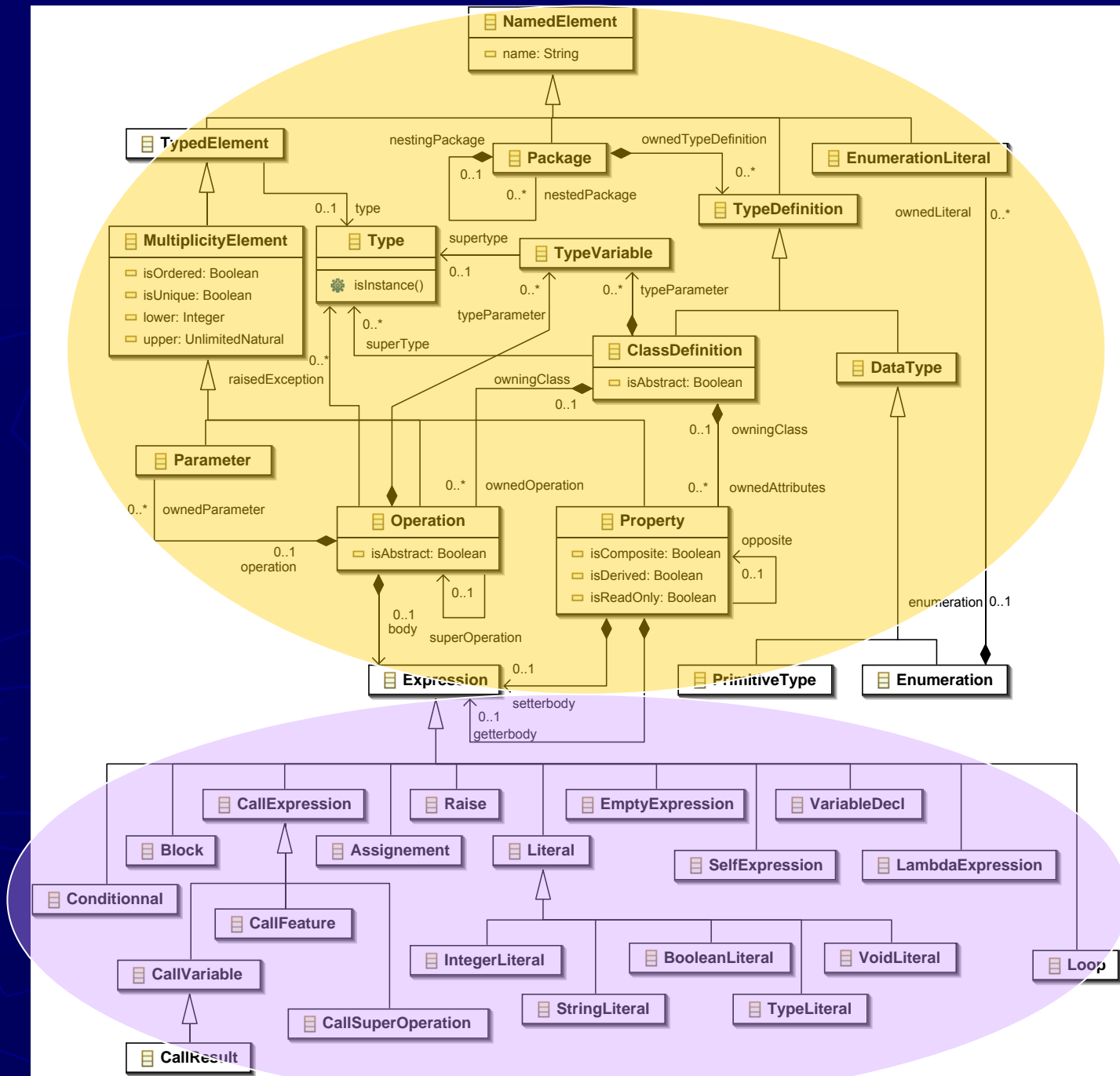


Construction par tissage d'aspect





KerMeta Metamodel



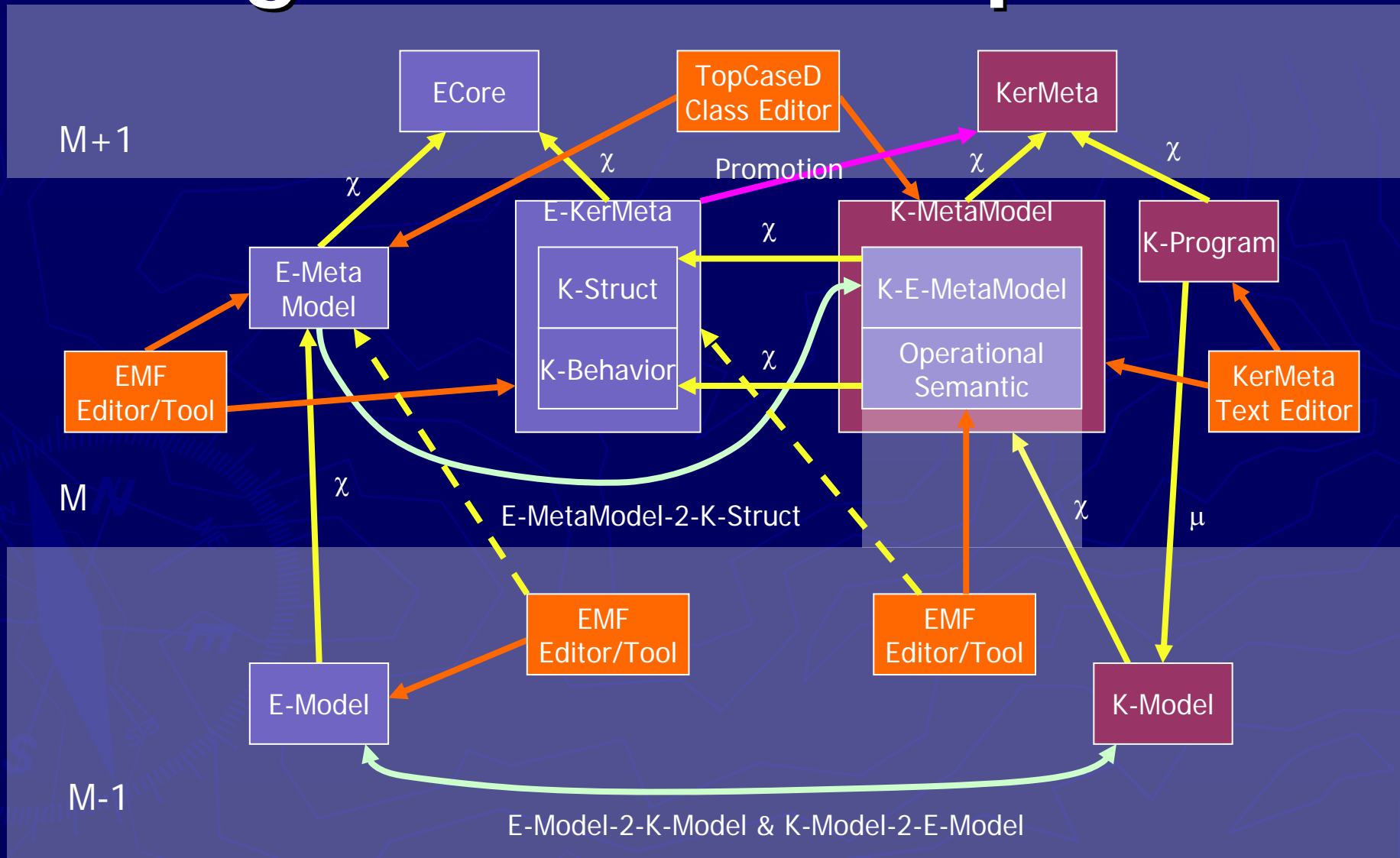


Intégration avec Eclipse/EMF





Integration avec Eclipse/EMF





Intégration avec Eclipse

Kermeta Metamodeling - fsm.km - Eclipse SDK

File Edit Navigate Search Project Density Run Window Help

100%

fr.irisa.triskell

```
result := validTransitions.one.fire
end

// Declaration of the post-condition
post notVoidOutput is
  result != void and result != ""

// Create a new state from self state
operation copy() : State is do
  result := State.new
  result.name := String.clone(name)
  result.combination := Set<State>.new
end

class Transition
{
  reference source : State[1..1]#outgoingTransition
  reference target : State[1..1]#incomingTransition
  attribute input : String
  attribute output : String

  // Fire the transition
  operation fire() : String is do
    // update FSM current state
    source.owningFSM.currentState := target
  result := output
  end
}
```

Resource Set

- platform:/resource/fr.irisa.triskell.kermeta.samples.fsm.
 - fsm
 - FSM
 - State
 - Transition
 - source from Transition
 - target from Transition
 - input from Transition
 - output from Transition
 - fire

Class -> String owned by fire

- Assignment false
- Call Feature currentState
- Call Feature owningFSM
- Call Feature target
- Assignment false
- Call Result result
- Call Feature output

FSMException

NonDeterminism

NoTransition

MetaModelStateFunction

Selection Parent List Tree Table Tree with Columns

fr.irisa.triskell.kermeta.samples.fsm.demo/kermeta/fsm.km

- Select
- Marquee
- Node
- Package
- ClassDefinition
- Property
- Operation
- Edge
- Inheritance link
- Reference

FSM

- ownedTransition : OrderedSet[0..1]
- alphabet : Set[0..1]
- run() : Void
- initialize(State, Boolean)

State

- name : String[0..1]
- combination : Set[0..1]
- step(String) : String
- copy() : State

Transition

- input : String[0..1]
- output : String[0..1]
- fire() : String

1 owningFSM

0..* ownedState

1 initialState

0..1 currentState

1 target

0..* outgoingTransition

1 source

0..* finalState

incomingTransition

0..*

Operation fire

Parameters Type Parameters Operation body

```
operation fire() : kermeta::standard::String is
do
  source.owningFSM.currentState := target
  result := output
end
```

OK Cancel

Console Properties Problems

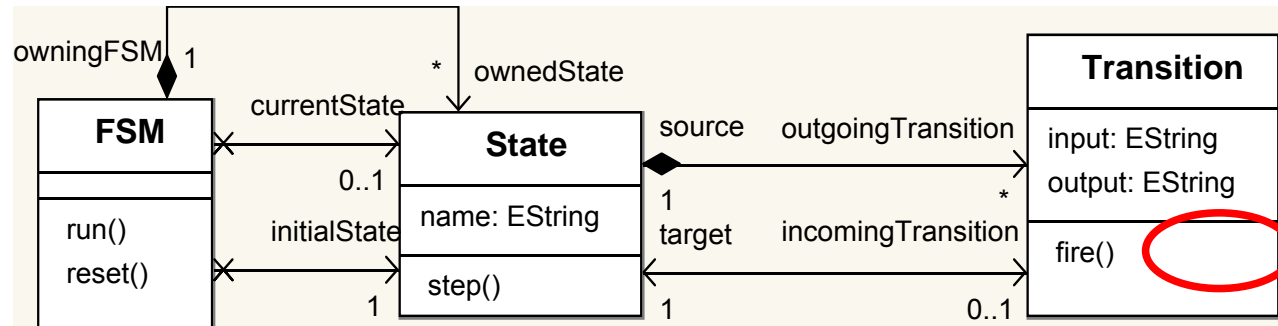
Graphics

Advanced

Property	Value
Is Abstract	false
Is Ordered	true
Is Unique	true
Lower	0
Name	fire
Raised Exception	

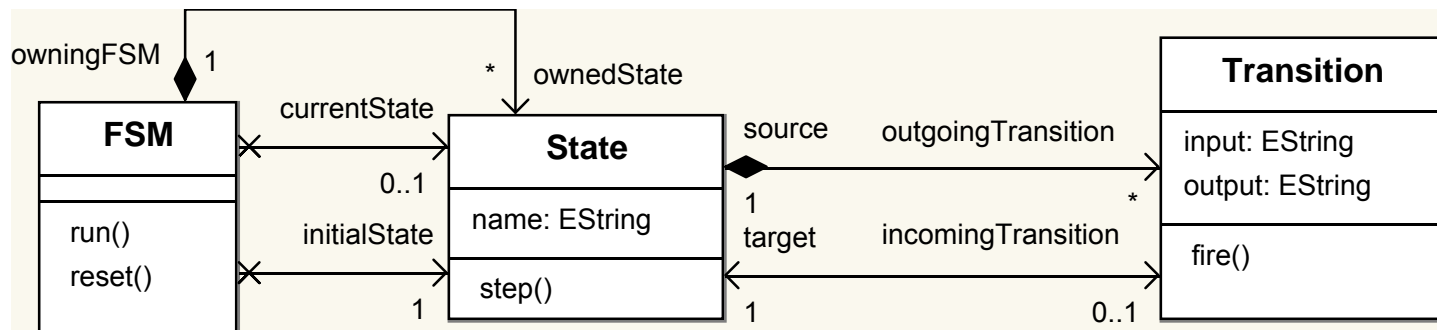


Example



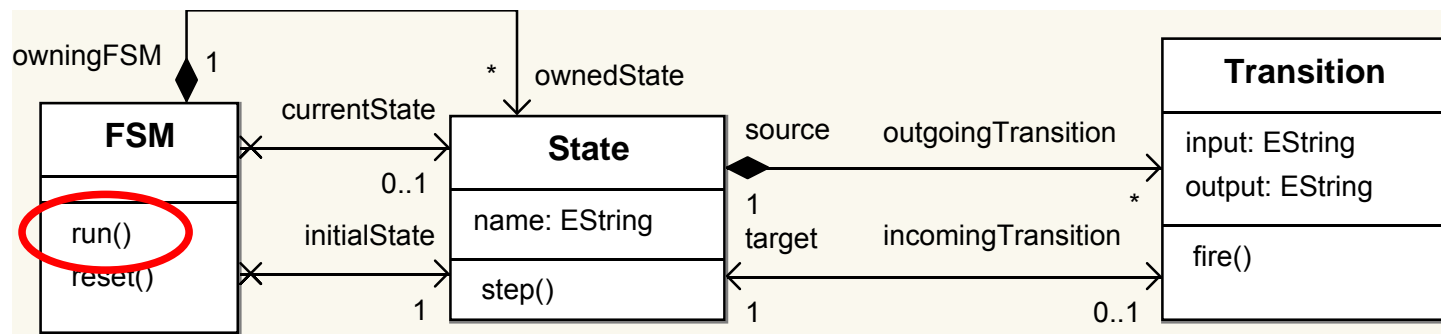
operation `fire() : String`

```
source.owningFSM.currentState := target  
result := output
```



operation `step(c : String) : String`

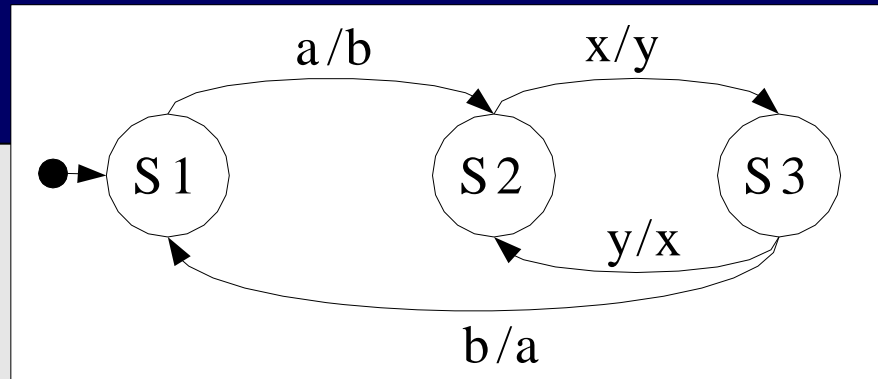
```
// Get the valid transitions
var validTransitions : Collection<Transition>
validTransitions := outgoingTransition.select { t |
    t.input.equals(c)
}
// Check if there is one and only one valid transition
if validTransitions.empty then raise NoTransition.new end
if validTransitions.size > 1 then
    raise NonDeterminism.new
end
// fire the transition
result := validTransitions.one.fire
```



operation run() : Void

```
from var str : String
until str == "exit"
loop
    stdio.writeln("current state is " + currentState.name)
    str := stdio.read("Enter an input string or 'exit'
                      to exit simulation : ")

    stdio.writeln(str)
    if str != "exit" then
        do
            stdio.writeln("Output string : " + currentState.step(str))
        rescue (ex : FSMException)
            stdio.writeln("ERROR : " + ex.toString)
        end
    end
end
end
stdio.writeln("** END OF SIMULATION **")
```



```
/**  
 * Create a sample FSM  
 */  
operation createFSM() : FSM is do  
 // The FSM  
 result := FSM.new  
  
 // Create the states of the FSM  
 var s1 : State init State.new          s1.name := "s1"          result.ownedState.add(s1)  
 var s2 : State init State.new          s2.name := "s2"          result.ownedState.add(s2)  
 var s3 : State init State.new          s3.name := "s3"          result.ownedState.add(s3)  
  
 // Create the transitions  
 var t12 : Transition init Transition.new  
 t12.input := "a"   t12.source := s1           t12.output := "b"   t12.target := s2  
 var t23x : Transition init Transition.new  
 t23x.input := "x"   t23x.source := s2   t23x.output := "y"   t23x.target := s3  
 var t23y : Transition init Transition.new  
 t23y.input := "y"   t23y.source := s2   t23y.output := "x"   t23y.target := s3  
 var t31 : Transition init Transition.new  
 t31.input := "b"   t31.source := s3           t31.output := "a"   t31.target := s1  
  
 // Set the initial state  
 result.initialState := s1  
end
```



Publication en annexe

- ▶ P.-A. Muller, F. Fleurey, et J.-M. Jézéquel. **Weaving executability into object-oriented meta-languages**. In *Proceedings of the MoDELS/UML 2005 - ACM/IEEE 8th International Conference on Model Driven Engineering*, pp. 264-278, November 2005. Lecture Notes in Computer Science, October 2005.

Langage et méthode pour une ingénierie des modèles fiable, soutenue le 9 octobre 2006, Université de Rennes



Metamodélisation exécutable

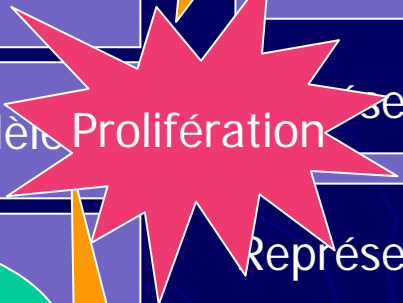
Système de metamodélisation

Syntaxes abstraite, et concrète, et sémantique

Comment décrire les metamodèle ?

Comment gérer l'explosion ?

Comment représenter un langage ?



Metamodèles métier

Système de metamodélisation

Opérationnalisation

Extrémités d'association

Démarche pour le déploiement d'UML

Comment fabriquer les choses ?

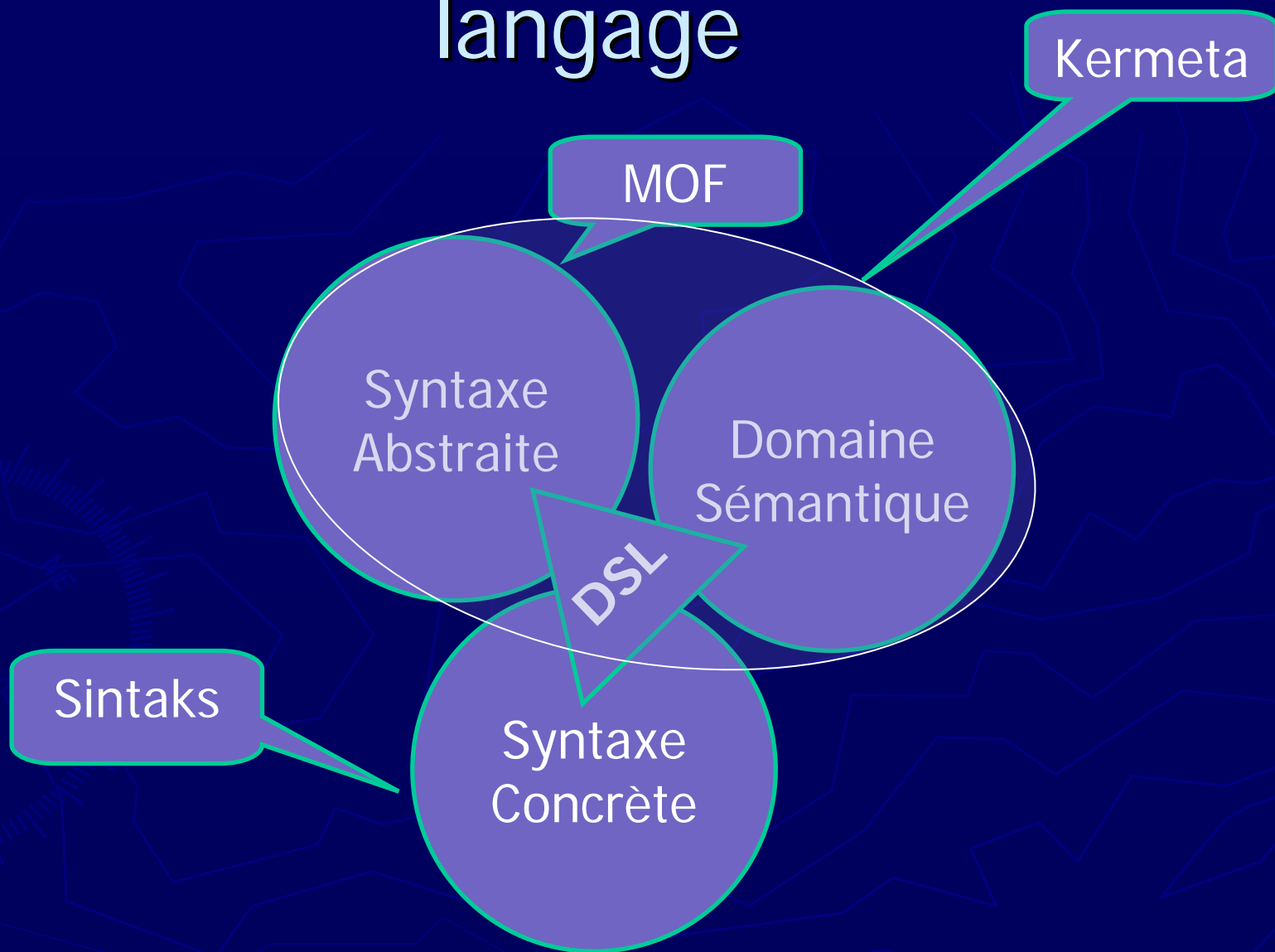
Sémantique des metamodèles

Comment utiliser UML ?

Comment intégrer tous ces langages ?



Retour sur la modélisation d'un langage





Motivation

▶ Comment créer des modèles conformes à des metamodels ?

- Par programme
- Avec un éditeur réflexif, éditeur graphique

Kermeta

EMF

TopCaseD

▶ Comment créer une représentation textuelle d'un modèle conforme à un metamodelle ?

- Syntaxe concrète générique
- Syntaxe concrète spécifique

HUTN

Lex/Flex
Yacc/Bison

▶ Grammarware : Parseur (text \rightarrow AST \rightarrow Model)ⁿ

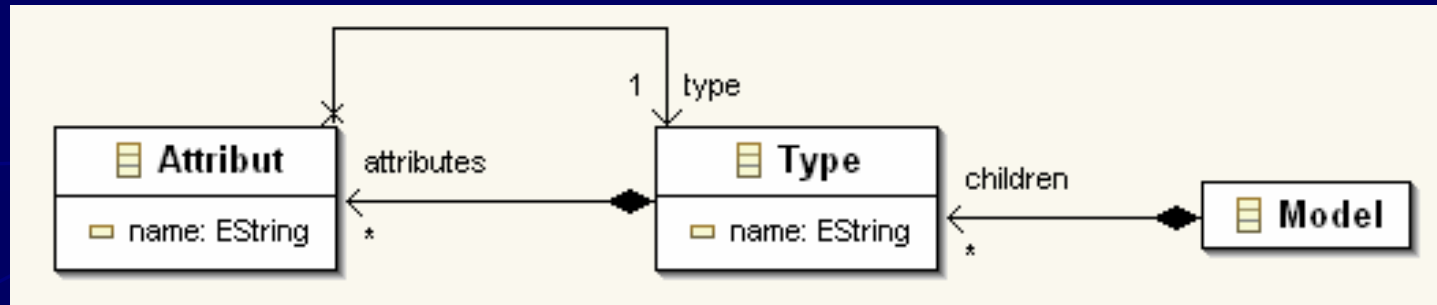
▶ Modelware : text \leftrightarrow Model

Sintaks



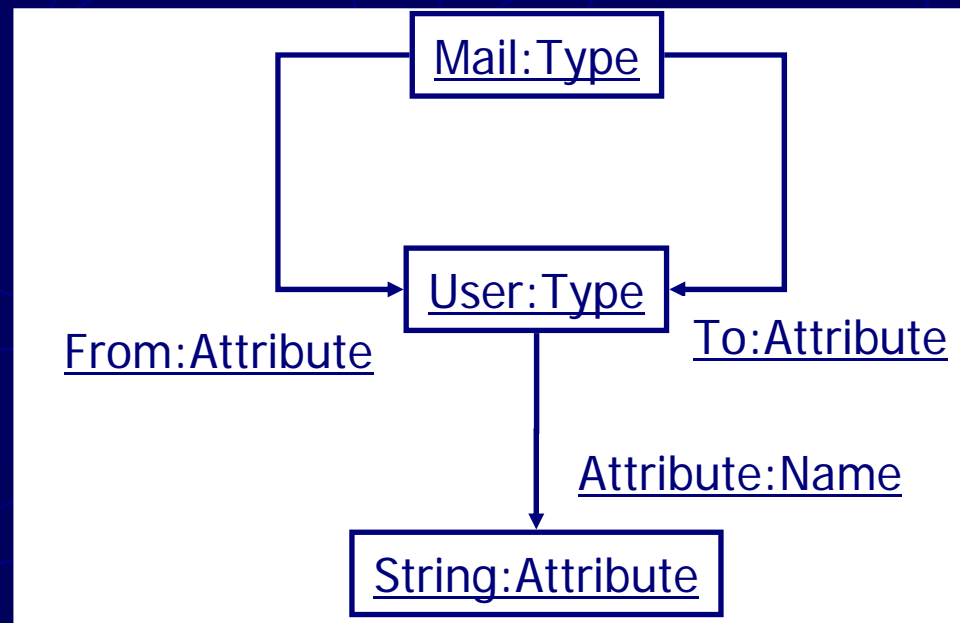
Exemple

▶ Petit metamodèle



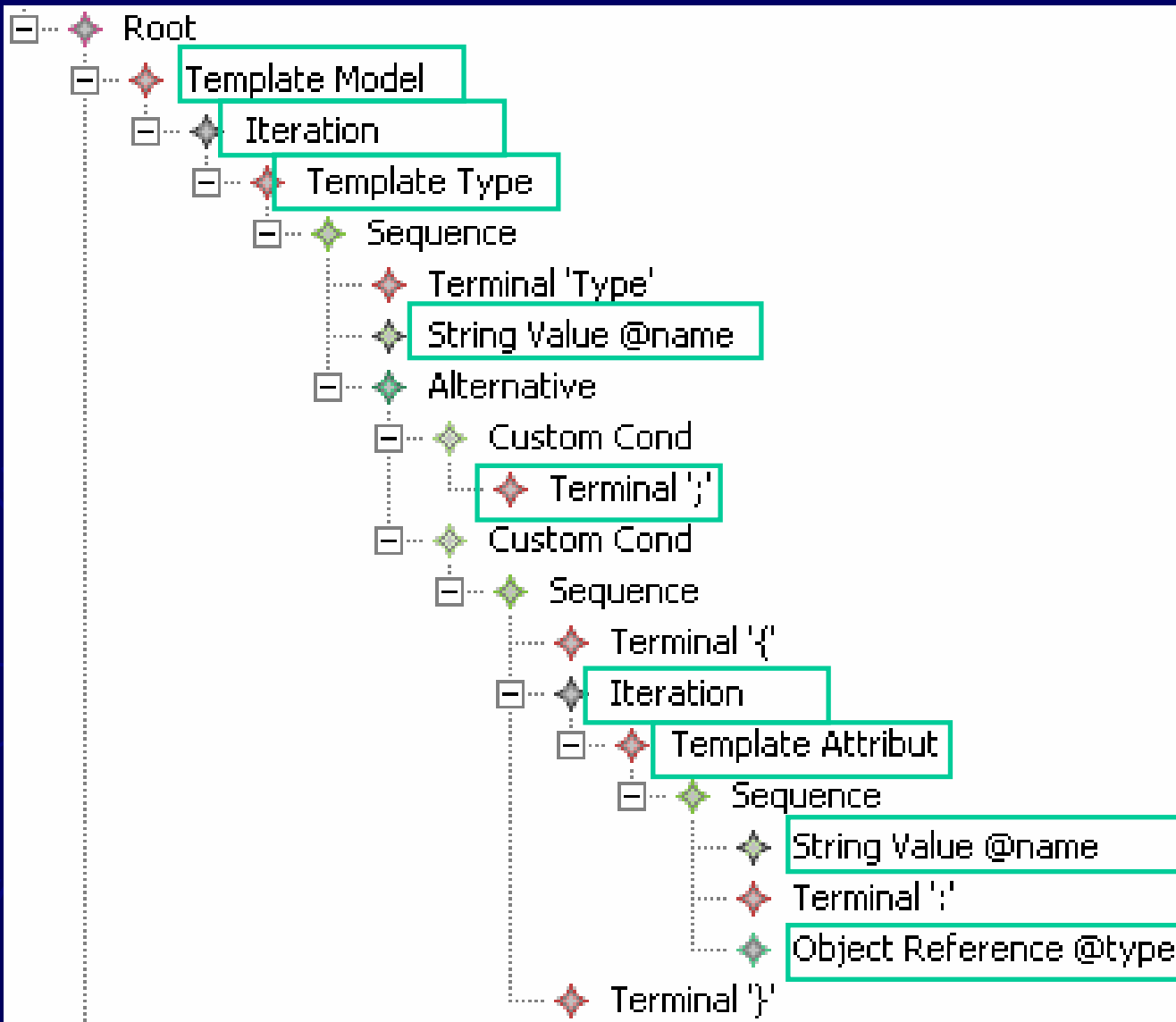
▶ Syntaxe concrète typique

```
Type Mail {
  From : User
  To : User
}
Type User {
  Name :
  String
}
Type String;
```





Modèle de syntaxe concrète



Working on Model

Fill a collection

Working on Type

Extract the type name

Just a semicolon

Fill a collection

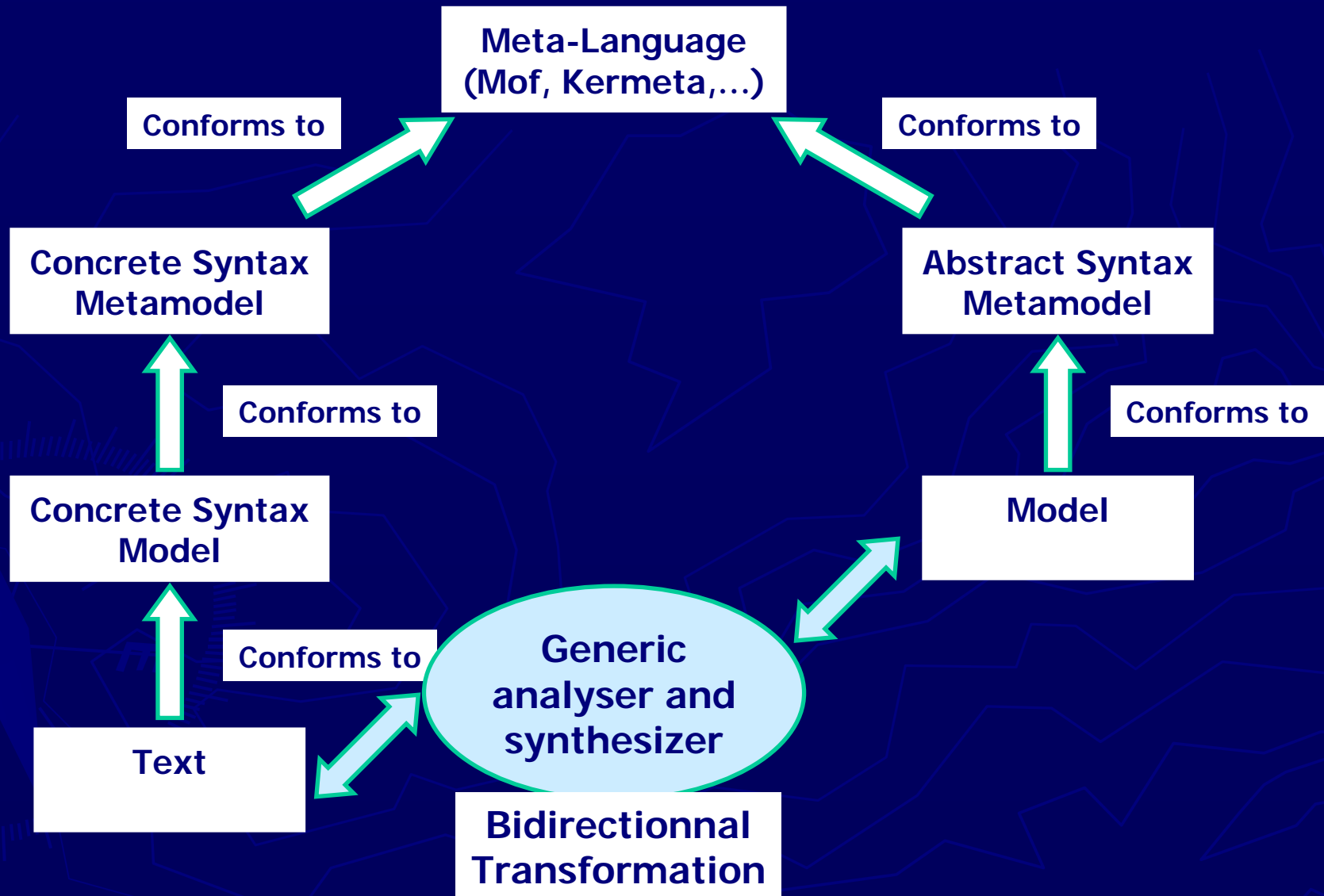
Working on Attribute

Extract the attribute name

Extract the ref to a type

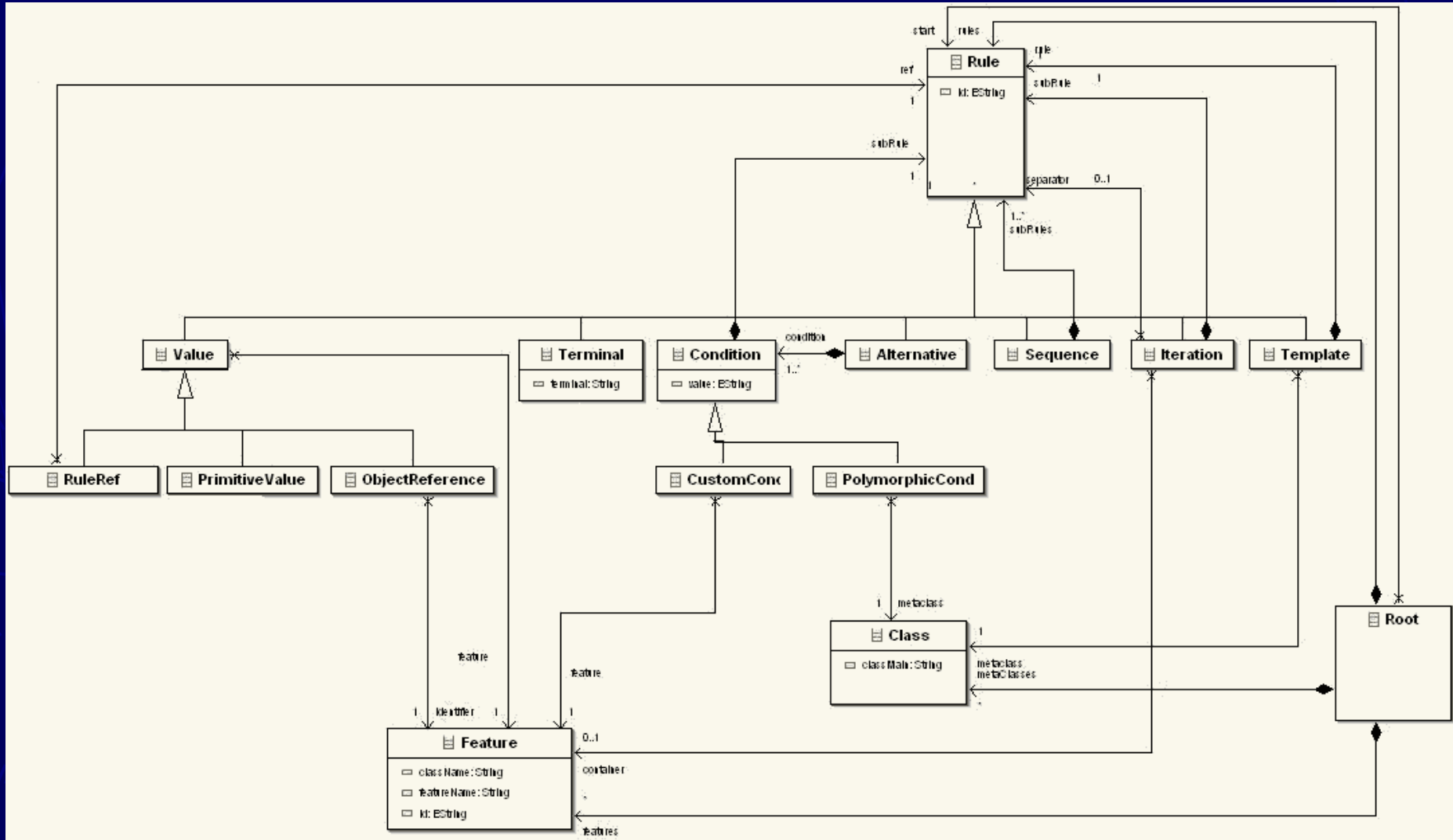


Principe général





Metamodelle de syntaxe concrète





Publication en annexe

- ▶ P.-A. Muller, F. Fleurey, F. Fondement, M. Hassenforder, R. Schneckenburger, S. Gérard, et J.-M. Jézéquel. **Model-Driven Analysis and Synthesis of Concrete Syntax.** In *Proceedings of the MoDELS/UML 2006 - ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems*, pp., Genova, Italy, Lecture Notes in Computer Science, 4199, October 2003.



Leçons apprises avec Kermeta

- ▶ Encore enrichir Kermeta pour en faire un vrai langage orienté-modèles
 - Fusion de metamodèles
 - Typage de modèles
- ▶ Changer de facteur d'échelle
 - Compilateur

Il faut se démarquer plus de EMF + Java





Perspectives



Et demain ?



Principaux verrous

- ▶ Comparaison des metamodèles, fusion et preuve d'équivalence
- ▶ Intégration de l'existant
- ▶ Quantification de l'impact sur les démarches





A court terme

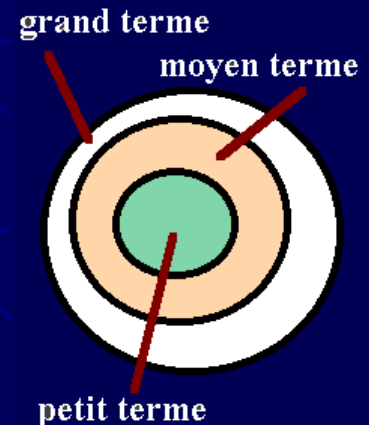
- ▶ Coordination de la plateforme RNTL
OpenEmbeDD
- ▶ Validation du passage à l'échelle de l'IDM
appliquée à l'ingénierie des langages





A moyen terme

- ▶ Projection vers des modèles formels et remontée de diagnostics
- ▶ Test des transformations de modèles
- ▶ Modélisation par aspects
- ▶ IDM, automatique et génie informatique
- ▶ Modélisation de l'intention
- ▶ Cartographie du génie-logiciel





Conclusion sur les contributions

- ▶ Des avancées méthodologiques
 - UML, IDM & Méthodes agiles
- ▶ Capitalisation de savoir-faire
 - Systèmes de commande-contrôle
- ▶ Modélisation opérationnelle
 - Systèmes d'information Web
- ▶ IDM pour l'ingénierie des langages
 - Kermeta & OpenEmbeDD







TOUR DU PROBLÈME IDM

C'EST VOTRE PREMIER TOUR DU PROBLÈME ?...

AH NON! JE L'AI DÉJÀ FAIT QUARANTE FOIS!

ET VOUS L'AVEZ GAGNÉ COMBIEN DE FOIS?

JAMAIS... ON NE GAGNE JAMAIS RIEN À FAIRE ÉTERNELLEMENT LE TOUR DU PROBLÈME !!!

C'EST CLAIR!

ET QUELLE SERA VOTRE TACTIQUE CETTE FOIS-CI?

CETTE FOIS-CI, JE VAIS PROCÉDER PAR ÉTAPES!

BIEN VU!

selon Mandryka 2005



Le Chat qui Pêche



2 rue des Francs Bourgeois

Ce soir à 19 heures



13 bis quai Châteaubriant